[ATUALIZADO] Análise: Android x86 Oreo (Parte I)

Easter egg da versão 8.x (exibido corretamente nas compilações do Android x86 Oreo).

Introdução

Desde o lançamento da versão 6 do Android para PCs e dispositivos com processador de arquitetura 32/64 bits, o qual foi analisado pelo blog, muita coisa aconteceu no projeto Android-x86, idealizado pelo engenheiro taiwanês Chih-Wei Huang, como o fim do projeto Remix OS, a sobrevivência do Phoenix OS, além do surgimento de novos projetos, como o OPENTHOS e o Prime OS (ambos com propostas semelhantes aos dois anteriores), além do Bliss OS (mais ambicioso que o próprio Android x86, com desenvolvimento mais amplo e lançamentos frequentes), sem esquecer de falar dos lançamentos estáveis da versão seguinte do SO do robozinho da Google, o já ultrapassado Nougat (7.x).

Mas, em 18 de Junho de 2018, o projeto começou a liberar compilações preliminares da versão Oreo (8.x) do Android que, apesar de não ter introduzido mudanças tão drásticas estruturalmente, possui diferenças significativas e aprimoramentos em relação ao sabor anterior, que serão objetos de análise deste artigo.

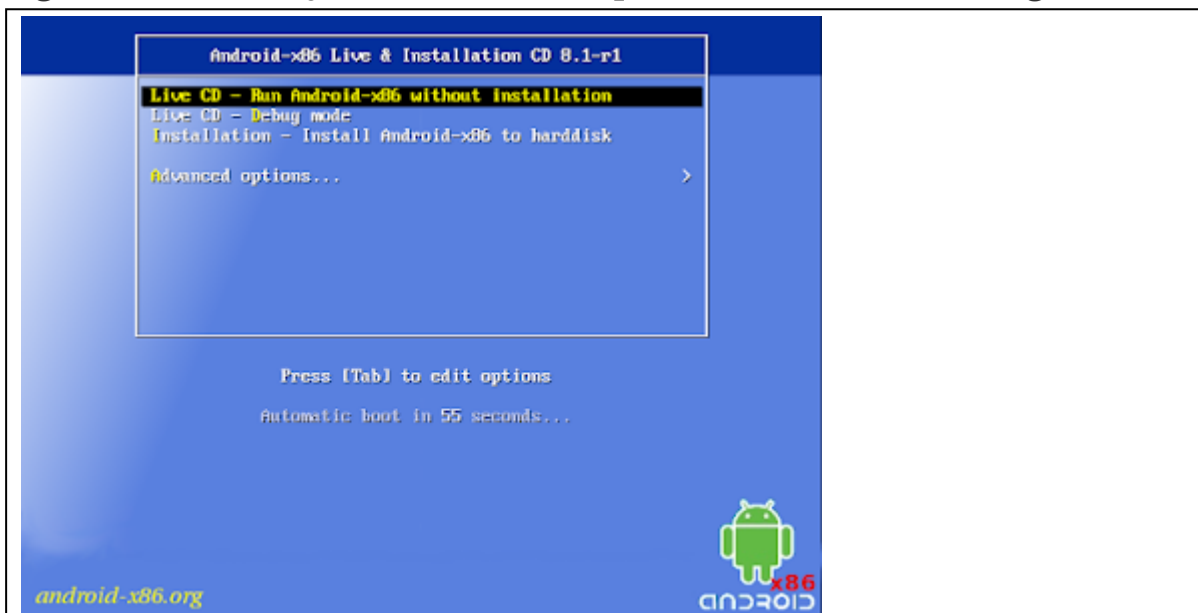
E, seis meses depois, em 15 de Janeiro de 2019, foi lançada a primeira versão estável, que pode ser baixada aqui ou aqui.

Agora, quase cinco meses após o primeiro lançamento, em 13 de Junho de 2019, veio oficialmente a segunda versão estável, cujas novidades serão detalhadas no final do segundo artigo, onde também será possível entender a razão de os links da versão anterior; no mais, para baixar a versão x86, x64 e x64 com kernel alternativo para usuários de máquinas virtuais, clique aqui, aqui ou aqui, respectivamente.

Já para o Nougat, para fins de comparação, foi utilizada a Release 2, segundo lançamento estável do projeto para esta edição, que pode ser baixada por aqui.

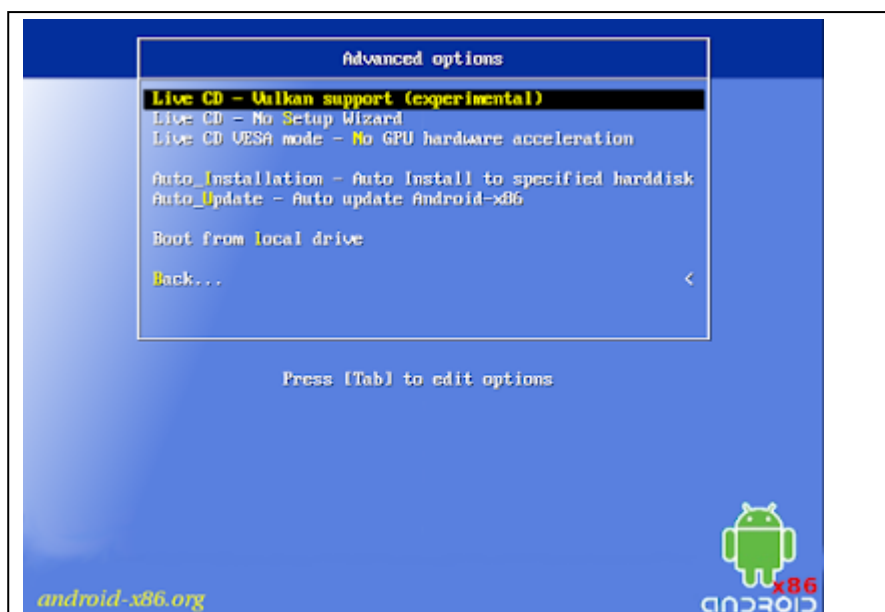
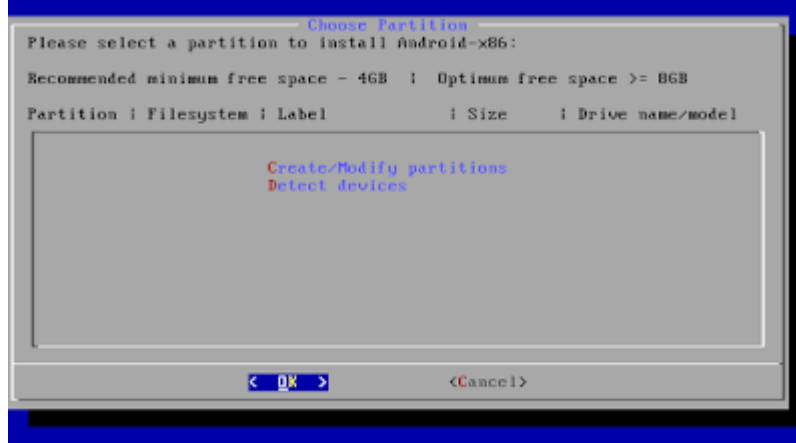
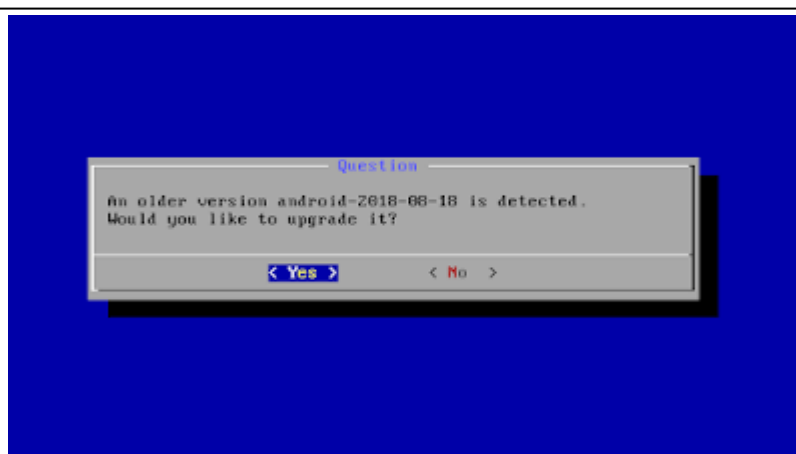
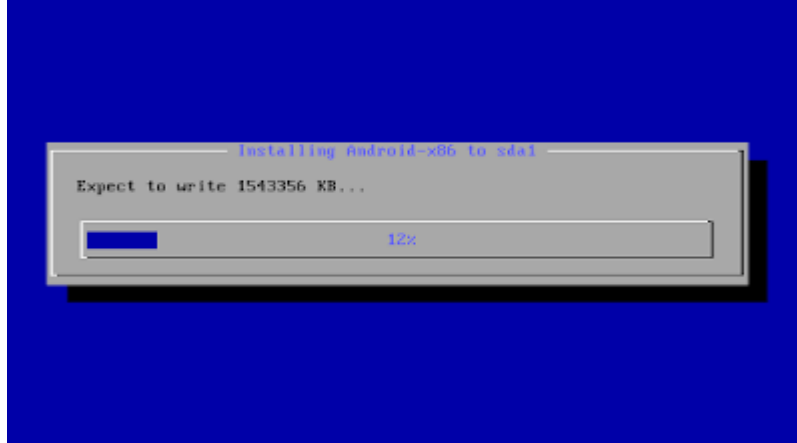
Para facilitar a leitura, a análise foi dividida em duas partes, onde aqui serão tratados dos seguintes temas:

Instalação

O guia de particionamento detalhado no artigo do Marshmallow ainda serve como base na instalação do sistema, mas o assistente sofreu algumas modificações desde então, que serão detalhadas a seguir.

Já na primeira tela, correspondente ao menu do Grub, foi adicionada a opção *Advanced Instalation*, que compreendem novos modos de

instalação do Android x86, detalhados na imagem a seguir.da

A opção Advanced Tools também foi aprimorada e agora conta com as seguintes opções:

- *Live CD - Vulkan Support* (opção que apareceu somente na primeira compilação estável, provavelmente em desenvolvimento para futuras compilações do projeto - considerando que o Android fornece suporte à biblioteca desde o Nougat);
- *Live CD - No Setup Wizard* (sem o assistente de configuração - modo que podia vir por padrão em determinadas compilações não oficiais, agora suportado pelo sistema);
- *Live CD VESA Mode* (movido da tela anterior, que pode ser útil em compilações não oficiais que não ajustam a tela automaticamente);
- *Auto_Installation* (que realiza o particionamento automático e outros procedimentos inerentes à instalação);
- *Auto_Update* (que facilita a instalação de atualizações do projeto, sem perder os aplicativos e as configurações feitas; novos modos de tela - entretanto, os arquivos atualizados serão instalados como somente leitura, mesmo que tenha sido instalado anteriormente como modo leitura-escrita, o que impede a remoção de aplicativos nativos, ainda que não afete os *apps* de usuário);
- *Inicializar do HD* (presente em muitas distros Linux, agora também no projeto).

Tela *Advanced Tools*, com as opções detalhadas acima.

Após o carregamento dos arquivos básicos, a tela de particionamento também sofreu modificações, com a exibição de novas informações e ficou um pouco mais coesa. Entretanto, em compilações não oficiais, pode haver um *delay* de até 20 segundos para tentar mover a seleção com o teclado.

Caso não queira utilizar a opção *Auto_Update* pelas limitações apontadas, também é possível fazer o *upgrade* no modo leitura-escrita, graças à opção implementada nas compilações oficiais mais recentes do Marshmallow, conforme informado naquela análise: pois bem, aqui o recurso foi aprimorado, permitindo, a partir do primeiro lançamento estável do Oreo, a atualização por cima de versões não oficiais do projeto (como mostra a imagem acima); para tal, é necessário selecionar a partição existente, NÃO reformatá-la, instalar o Grub e confirmar a instalação no modo leitura-escrita, para que, então, o assistente identifique a existência da instalação anterior, dando a opção de preservar os dados da partição */data*, como ocorre nos dispositivos móveis com o Android embarcado (vale ressaltar que, aqui, a funcionalidade suporta o upgrade de um sistema x86 para x64 e vice-versa, considerando que o processador seja 64 bits, embora não seja recomendado por causa do problema conhecido envolvendo o Google Play Services, que será relatado mais adiante).

No mais, seguindo o padrão de tamanho das compilações anteriores, a versão x86 do Oreo R1 consome cerca de 1,48 GB de espaço no modo leitura-escrita (650 MB no modo somente leitura).

```
[    4.0533341] EXT4-fs (loop0): couldn't mount as ext3 due to feature incompatib
ilities
[    4.1305731] EXT4-fs (loop0): mounted filesystem with ordered data mode. Opts:
 (null)
 found at /dev/sda1
[    4.5134241] hrtimer: interrupt took 3697595 ns
[    4.7531401] random: crng init done
[    4.7548091] random: 7 urandom warning(s) missed due to ratelimiting
[    5.4179941] input: AT Translated Set 2 keyboard as /devices/platform/i8042/se
rio0/input/input4

Type 'exit' to continue booting...

Running MirBSD Korn Shell...
android:/android #
android:/android # exit
[   48.7528611] EXT4-fs (sda1): re-mounted. Opts: (null)

Use Alt-F1/F2/F3 to switch between virtual consoles
Type 'exit' to enter Android...

Running MirBSD Korn Shell...
android:/android # mount -o remount,rw /mnt
[   59.9782171] EXT4-fs (sda1): re-mounted. Opts: data=ordered
android:/android #
```

Antes de partir para a próxima seção, dependendo das configurações da máquina (seja física ou hospedeira de uma máquina virtual), o sistema poderá sofrer *bootloop* (repetição intermitente da tela de *boot*, bastante familiar para usuários de *smartphone* Android), por causa de problema com o driver de vídeo; para atenuar isto de forma permanente, é necessário editar o Grub do sistema, selecionando a opção de depuração (*debug mode*) e, então, digitar 'exit' para continuar a operação (se necessário, pode-se alternar para outro console vazio usando ALT + F2), em seguida, digitar o comando *mount -o remount,rw /mnt* (caso o teclado do dispositivo não corresponda com o mapa do teclado internacional, no caso do PT-BR ABNT2, o '/' será gerado pelo botão ';').

```
/mnt/grub # cd /
/ # ls
android  dev      init     lib      proc     sbin     sfs      sys      tmp
bin      hd       iso      mnt      root     scripts  src      system
/ # cd mnt
/mnt # ls
android-7.1-r2  grub             lost+found
/mnt # cd grub
/mnt/grub # ls
android-x86.xpm.gz  fat_stage1_5        stage1
device.map          menu.lst            stage2
e2fs_stage1_5       ntfs_stage1_5       stage2_eltorito
/mnt/grub #
```

Após checar se o dispositivo foi montado corretamente (conforme é mostrado na imagem acima), digitar o comando *vi /mnt/grub/menu.lst* para editar o arquivo de configuração da inicialização do sistema.

```
default=0
timeout=6
splashimage=/grub/android-x86.xpm.gz
root (hd0,0)

title Android-x86 7.1-r2
        kernel /android-7.1-r2/kernel nomodeset quiet root=/dev/ram0 androidboot
        initrd /android-7.1-r2/initrd.img

title Android-x86 7.1-r2 (Debug mode)
        kernel /android-7.1-r2/kernel root=/dev/ram0 androidboot.selinux=permiss
        initrd /android-7.1-r2/initrd.img

title Android-x86 7.1-r2 (Debug nomodeset)
        kernel /android-7.1-r2/kernel nomodeset root=/dev/ram0 androidboot.selin
        initrd /android-7.1-r2/initrd.img

title Android-x86 7.1-r2 (Debug video=LVDS-1:d)
        kernel /android-7.1-r2/kernel video=LVDS-1:d root=/dev/ram0 androidboot.
        initrd /android-7.1-r2/initrd.img
~
~
~
:wq_
```

Dentro do editor, posicionar o cursor após a palavra 'kernel' e apertar SHIFT + V para habilitar a edição; então deve-se digitar *nomodeset* antes de 'quiet'; em seguida, clica-se em ESC para sair do modo de edição e, para salvar as mudanças e sair do editor, digite o comando :wq (caso o teclado do dispositivo não corresponda com o mapa do teclado internacional, no caso do PT-BR ABNT2, o ':' será gerado pelo botão 'Ç').

```
  Trusted GRUB 1.1.5 (http://trustedgrub.sf.net)
  [ No TPM detected! ] (639K lower / 1047488K upper memory)
                                                                [0]
 kernel /android-7.1-r2/kernel nomodeset quiet root=/dev/ram0 androidb+
 initrd /android-7.1-r2/initrd.img

    Press 'b' to boot, 'e' to edit the selected command in the
       boot sequence, 'c' for a command-line, 'o' to open a new line
       after ('O' for before) the selected line, 'd' to remove the
       selected line, '/?nN' to search, or escape to go back to the main menu.
```

Em seguida, digitar o comando *sync* e, por fim, *reboot -f* para forçar o encerramento de operações e reinicializar a máquina. Assim, o comando será exibido como acima quando o Grub for carregado novamente e o sistema deverá iniciar a interface sem grandes problemas.

Uso do Swap

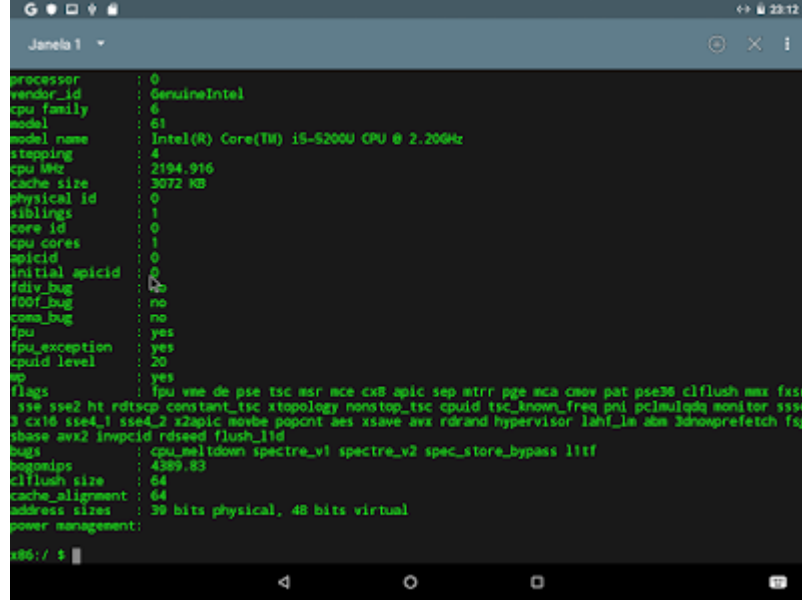
Antes de prosseguir, estarei considerando que você tenha conhecimentos sobre o conceito de SWAP (área de troca), já abordado no artigo sobre o Marshmallow, mas que aqui será abordado de outra forma, e poderá servir de referência para uso em futuras versões do Android x86 ou mesmo em distribuições GNU/Linux em geral.

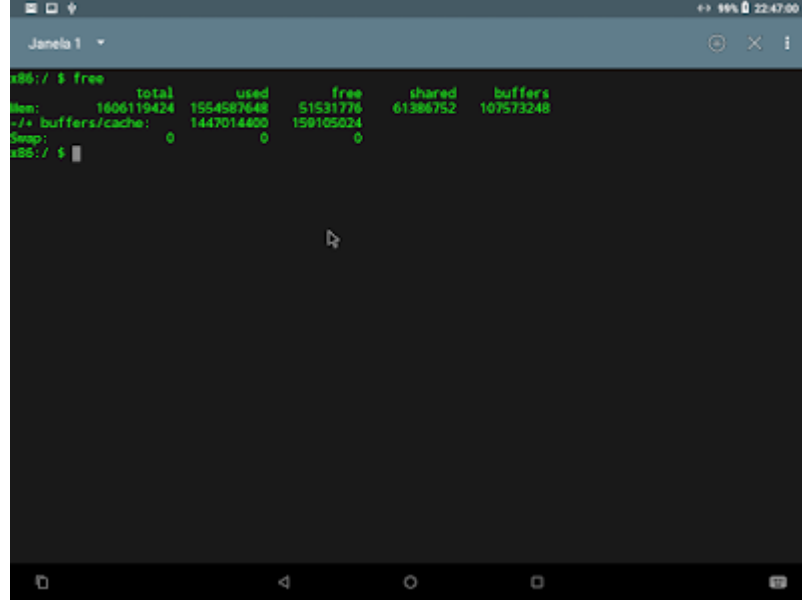
Considerando a tendência do Ubuntu em adotar a instalação automática de arquivo *swap* (a partir do 17.04, como fala aqui), em substituição à partição de área de troca, pela sua maleabilidade e funcionamento em geral (que o torna mais próximo da memória virtual e hibernação do Windows), aprimorado nas versões mais recentes do kernel Linux –

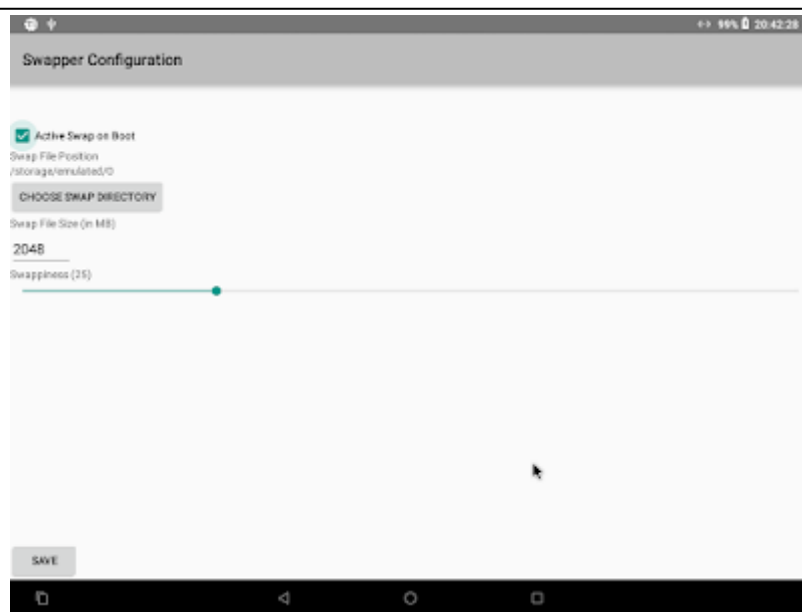
vale ressaltar que a funcionalidade do arquivo *swap* não é, necessariamente, um recurso novo no kernel criado por Linus Torvalds, como poderá ser constatado mais adiante - pensei em testá-lo para esta análise.

Para tal, levei em consideração tutoriais antigos (este e este, por exemplo), que abordavam seu uso na época do Android Gingerbread (2.3.x; SDK 9), onde os dispositivos embarcados com esta versão geralmente tinham hardware mais limitado, em especial, com menos memória RAM (em comparação com os modelos mais atuais) onde, com aplicativos que já exigiam bastantes recursos do sistema, era uma alternativa para aprimorar um pouco o funcionamento do aparelho.

Os procedimentos a seguir exigem permissão de super usuário (ou seja, o dispositivo precisa estar "rooteado", requisito que o Android x86 atende por padrão), pois lida com arquivos e processos que afetam o sistema.

- Primeiro, acesse a gaveta de aplicativos do sistema e localize o Emulador de Terminal;
- Digite o comando **su** e permita a execução do modo super usuário (caso tenha aberto o *app* pela primeira vez);
- **dd if=/dev/zero of=/mnt/swapfile bs=1024 count=2097152** (para criar o arquivo *swapfile*, no caminho */mnt*, com 2097152 blocos de 1 KB cada - mude a quantidade de blocos de acordo com a necessidade, mas tome cuidado para não aumentar demais o tamanho do bloco, que segue a mesma ideia de setor de um sistema de arquivos qualquer);
- **chmod 600 /swapfile** (para configurar a permissão do arquivo, onde só o usuário *root* tem acesso de leitura e escrita nele);
- **losetup /dev/block/loop0 /mnt/swapfile** (opcional, para associar o arquivo num dispositivo de repetição, a fim de facilitar sua inicialização automática, já que o Android não conta com o CRON, por exemplo);
- **mkswap /dev/block/loop0** ou **mkswap /mnt/swapfile** (a fim de tornar o arquivo criado como Swap propriamente);
- **swapon /dev/block/loop0** ou **swapon /mnt/swapfile** (para inicializar o Swap no sistema).

Em meus testes, até o penúltimo comando ocorreu sem problemas (podendo ter cenários onde até o *mkswap* falhe, impedindo a inicialização do sistema), mas, como mostra a imagem acima, ao tentar ativar a área de troca, dá um erro que, em tese, não era para acontecer.

Ao checar os detalhes do comando [swapon](), percebe-se que ele é uma versão simplificada do que originalmente é encontrado nas distribuições GNU/Linux.

```
Janela 1
processor       : 0
vendor_id       : GenuineIntel
cpu family      : 6
model           : 61
model name      : Intel(R) Core(TM) i5-5200U CPU @ 2.20GHz
stepping        : 4
cpu MHz         : 2194.916
cache size      : 3072 KB
physical id     : 0
siblings        : 1
core id         : 0
cpu cores       : 1
apicid          : 0
initial apicid  : 0
fdiv_bug        : no
f00f_bug        : no
coma_bug        : no
fpu             : yes
fpu_exception   : yes
cpuid level     : 20
wp              : yes
flags           : fpu vme de pse tsc msr mce cx8 apic sep mtrr pge mca cmov pat pse36 clflush mmx fxsr
 sse sse2 ht rdtscp constant_tsc xtopology nonstop_tsc cpuid tsc_known_freq pni pclmulqdq monitor ssse
3 cx16 sse4_1 sse4_2 x2apic movbe popcnt aes xsave avx rdrand hypervisor lahf_lm abm 3dnowprefetch fsg
sbase avx2 invpcid rdseed flush_l1d
bugs            : cpu_meltdown spectre_v1 spectre_v2 spec_store_bypass l1tf
bogomips        : 4389.83
clflush size    : 64
cache_alignment : 64
address sizes   : 39 bits physical, 48 bits virtual
power management:

x86:/ $
```

Então, a partir de um comando *cat /proc/cpuinfo* (indicado no artigo do segundo link acima), a fim de verificar se o kernel possui o recurso *swp* ou *swap*, constatei que a exibição do arquivo segue um padrão um pouco diferente daquele tutorial e o recurso, em si, aparentemente, não está listado - o que sugere que o kernel embarcado nas compilações do Android x86 não dá suporte ao recurso.

A partir do mesmo artigo, ainda tentei utilizar o aplicativo Swapper & Tools, que automatiza todo o processo, inclusive permitindo a configuração do valor do *swappiness*, que, basicamente, controla quando o *swap* pode ser utilizado.

No final de contas, ao checar o comando *free*, que mostra o consumo de memória física, mostra valores zerados para o *swap*, sugerindo que o recurso não está sendo utilizado, mesmo após ter sido configurado.

Em resumo, é provável que o Android x86 não dê suporte ao recurso (pelo menos não, por enquanto), o que é uma pena, já que é uma alternativa a menos para aprimorar o funcionamento do sistema (e que, teoricamente, o SO da Google deve suportar já há muito tempo).

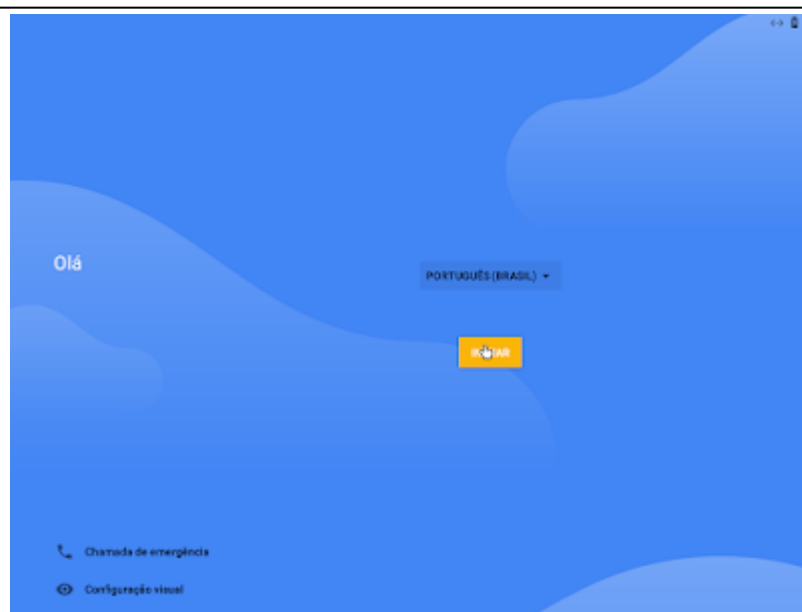
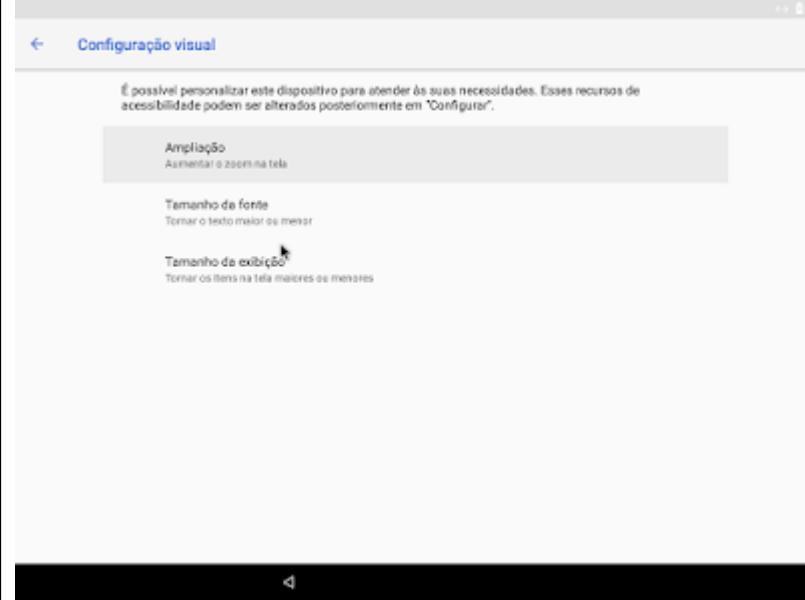
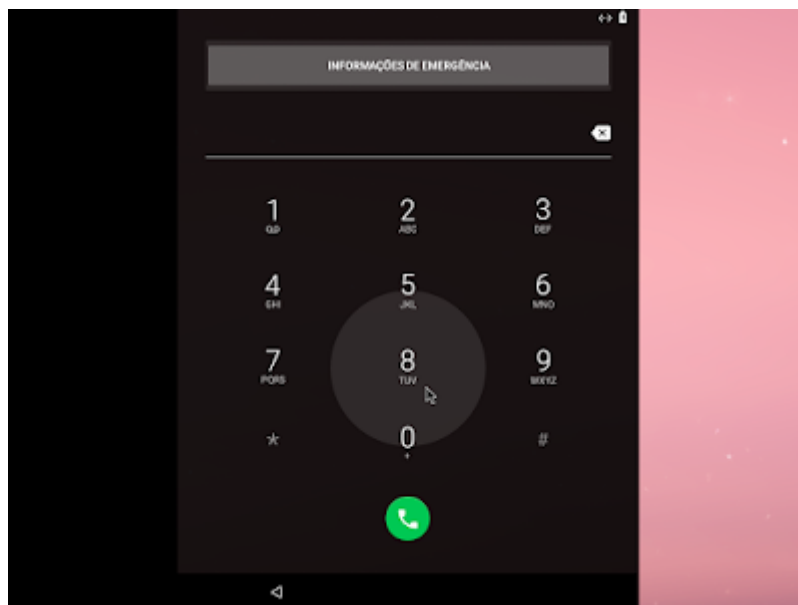
Assistente de configuração

O assistente sofreu modificações em relação às versões Lollipop/Marshmallow e, por esta razão, serão detalhadas a seguir, para fins de comparação (também é válido para o Nougat):

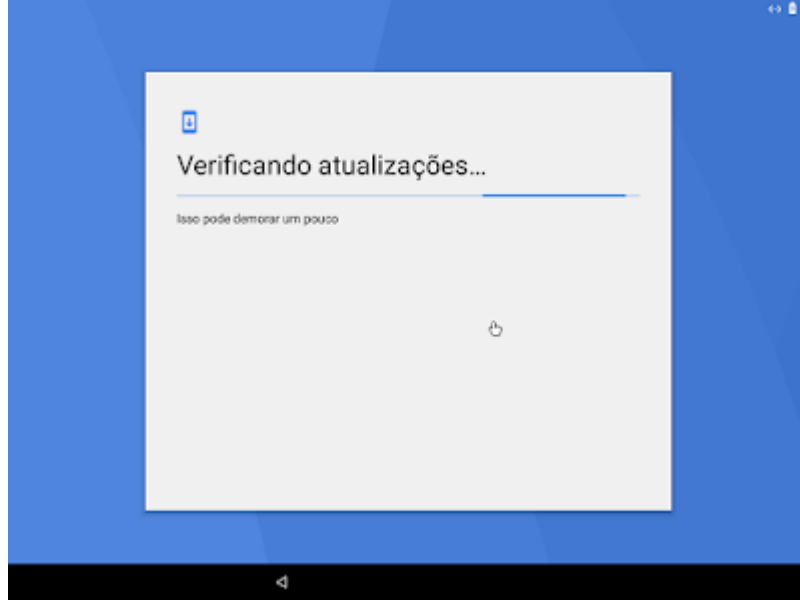
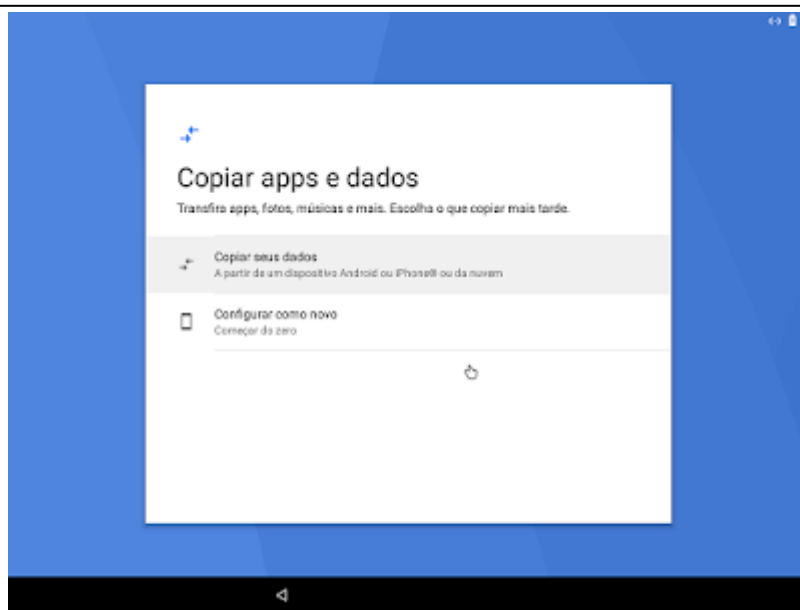
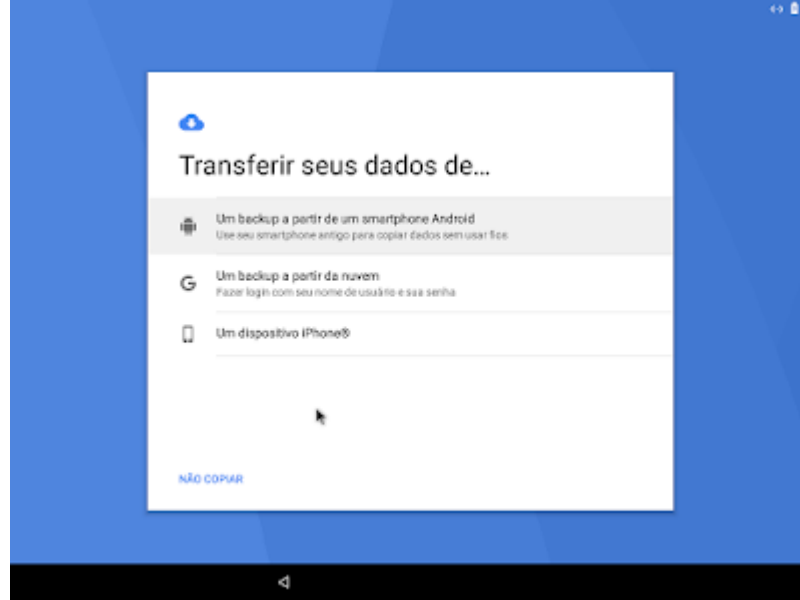
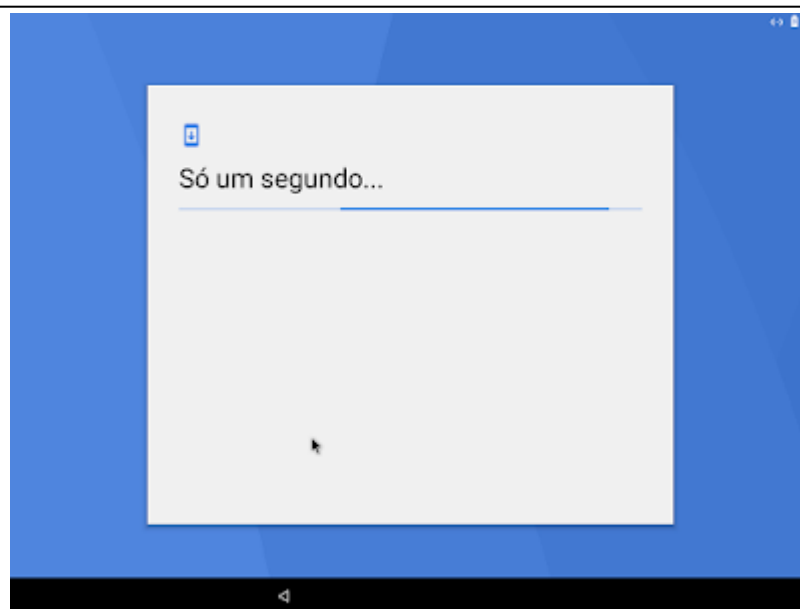
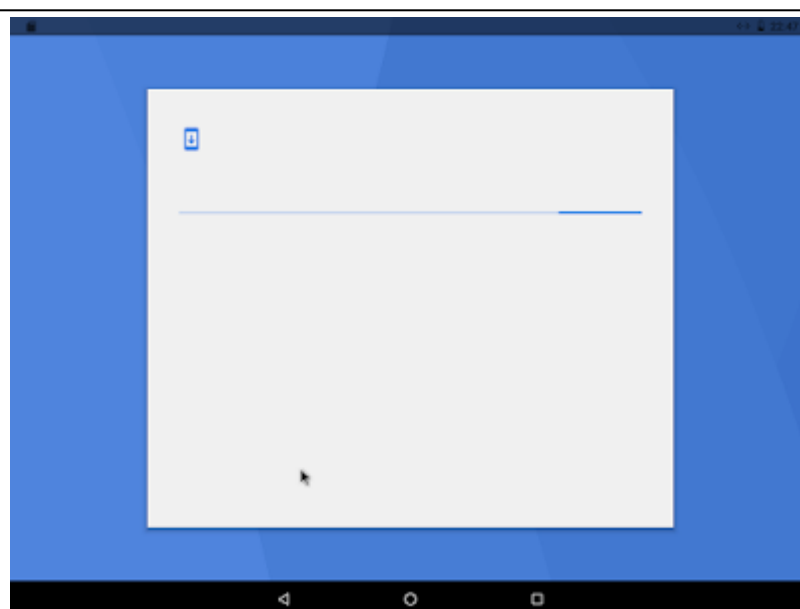
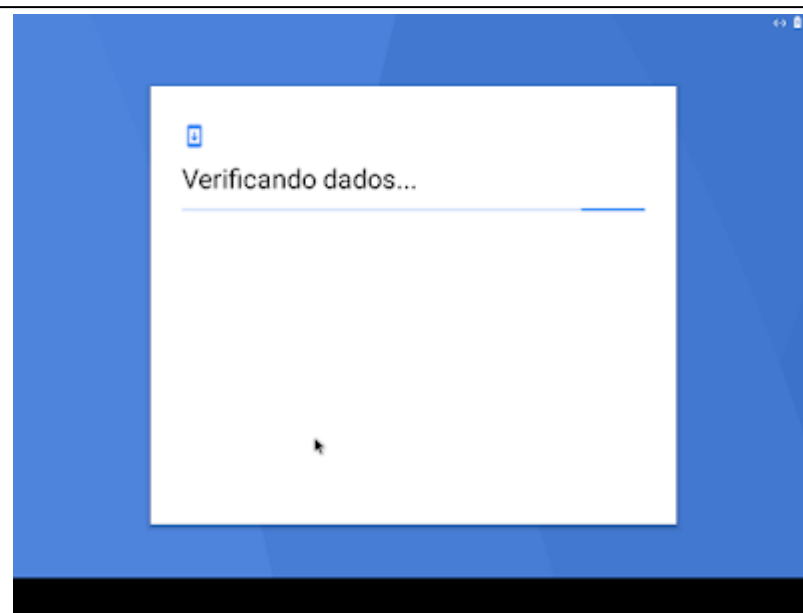
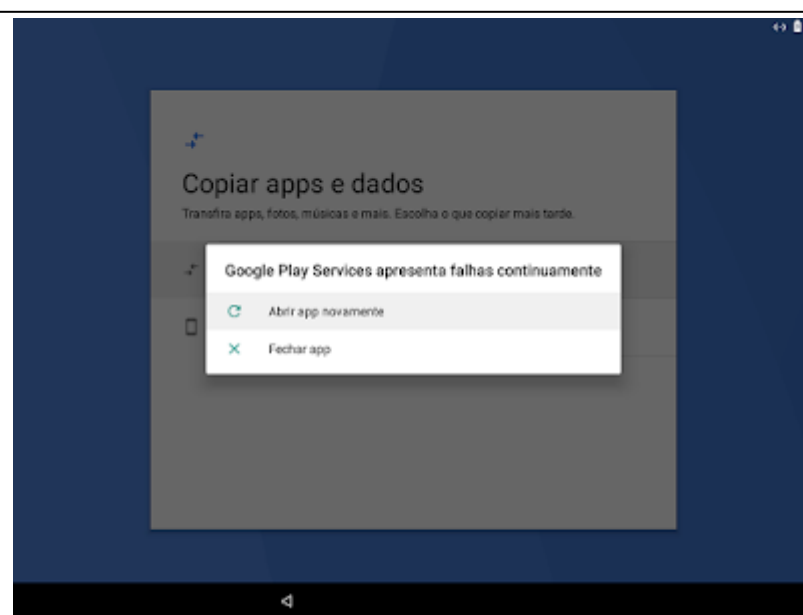
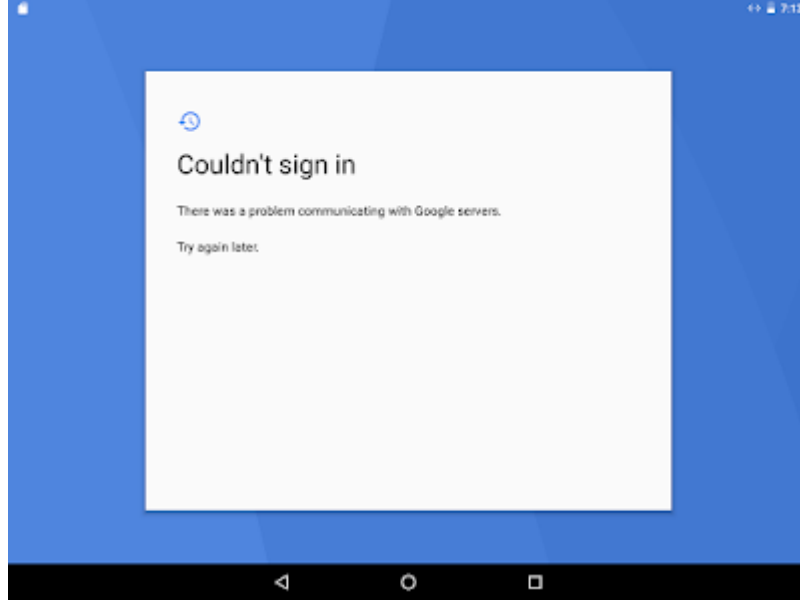
Antes de tudo, ao iniciar a instalação em modo gráfico, o usuário vai perceber que o ponteiro do mouse aparece automaticamente, não exigindo mais desabilitar a integração do mouse, no menu Entrada do VirtualBox, o que foi uma adição do projeto a partir da primeira versão estável.

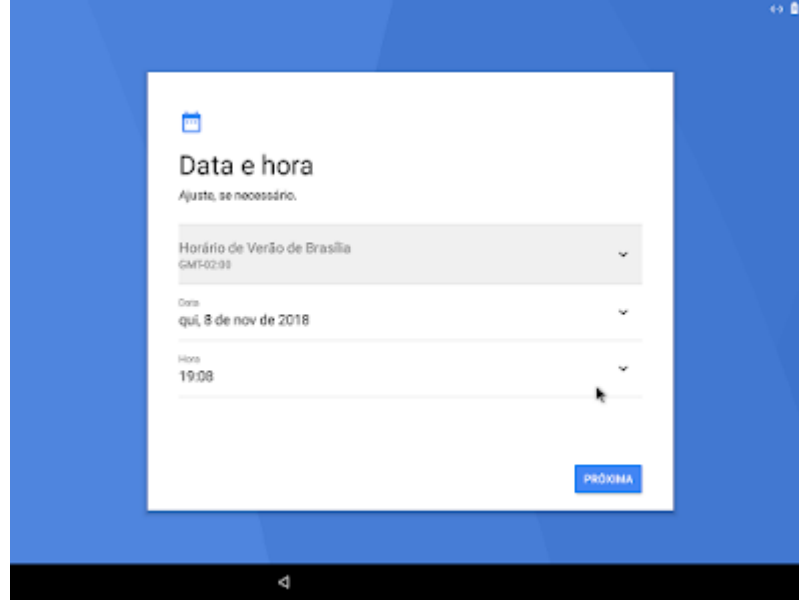
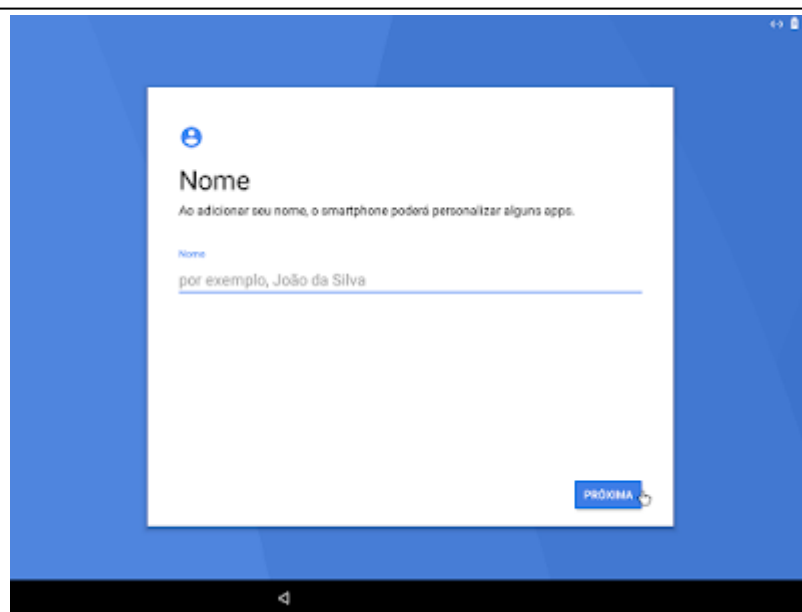
Nas compilações do Oreo testadas para o artigo (incluindo as preliminares), em relação às estáveis do Nougat, logo ao começar o assistente de configuração, deu para perceber que o sistema, pelo menos na versão 32 bits, ainda está instável, com frequentes falhas no processo *system*, mas que não compromete totalmente o uso, dando para ignorar os avisos, mas pode ser necessária alguma dose de paciência.

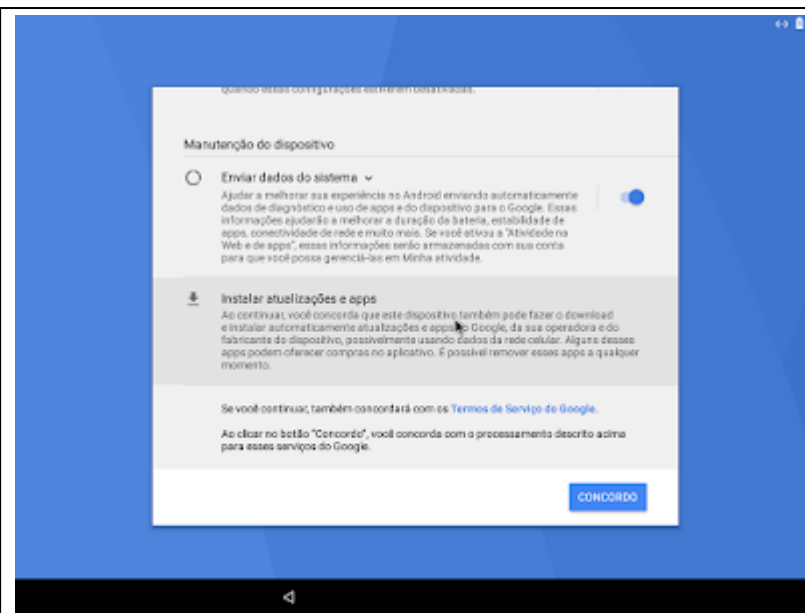
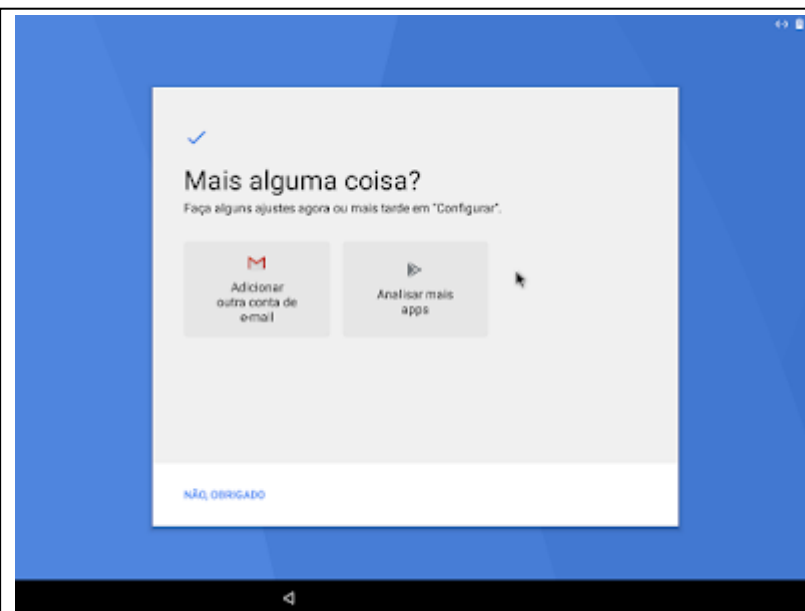
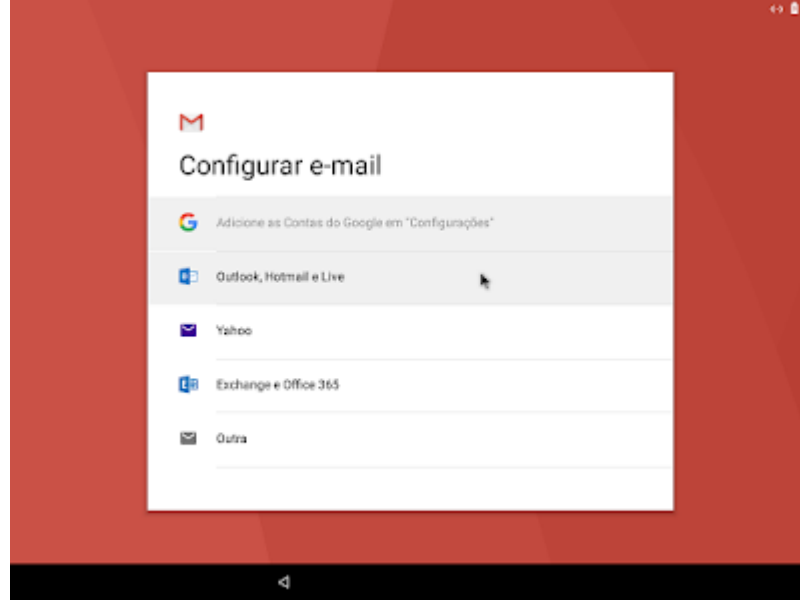
A seleção de idiomas também sofreu modificações: apesar da sugestão dos idiomas Inglês (EUA) e espanhol, estes passaram a ficar agrupados e, para selecionar a linguagem PT-BR, agora são necessários mais passos: primeiro, selecione a opção 'Português'.

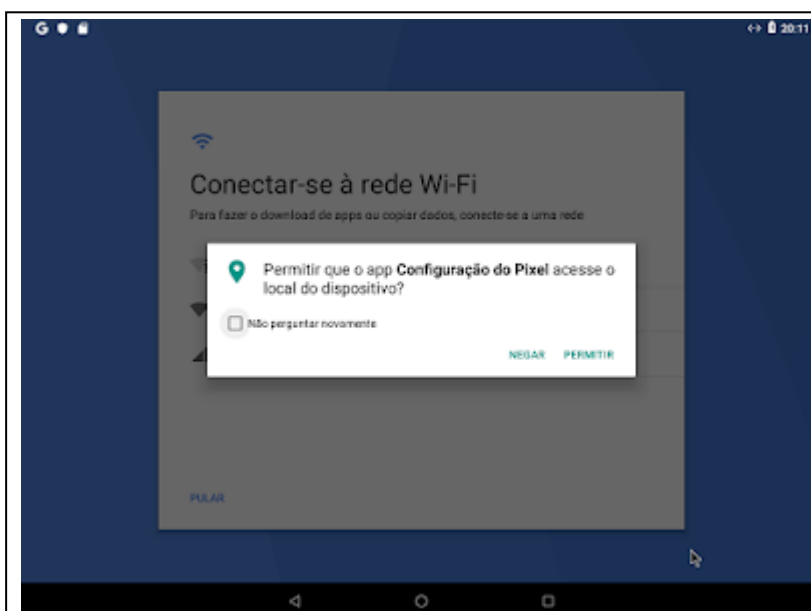
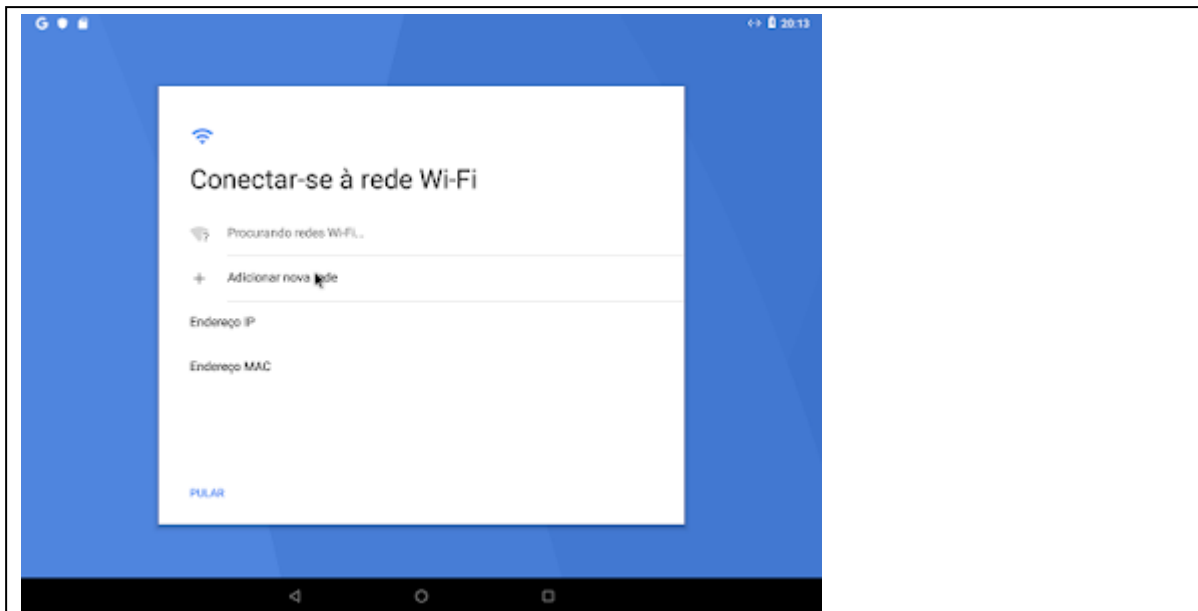
Em seguida, selecione o país correspondente - no caso, Brasil e Portugal são os sugeridos, por ser os países principais. E o ajuste é feito imediatamente.

As diferenças na tela de boas-vindas são o fundo diferente, na saudação diferente (antes “Bem-vindo”, agora “Olá”); no botão (antes "Vamos lá" e cor azul; agora "Iniciar" e cor amarela) e também a opção Configuração Visual (que será detalhado a seguir).

Clicando em Configuração Visual, são abertas as opções que também podem ser acessadas em Configurações - Tela e que, por esta razão, serão detalhadas no tópico correspondente. Vale ressaltar que, apesar de ser a mesma tela no Nougat, percebe-se, de cara, a inversão da cor na barra de título, o que deixou o Android ainda mais branco do que se tornou desde o Lollipop (ainda que haja um atenuante no Oreo que será falado mais adiante).

Já as Chamadas de Emergência, pouco se alterou desde o Lollipop, incluindo ainda não estar adaptado ao modo paisagem.

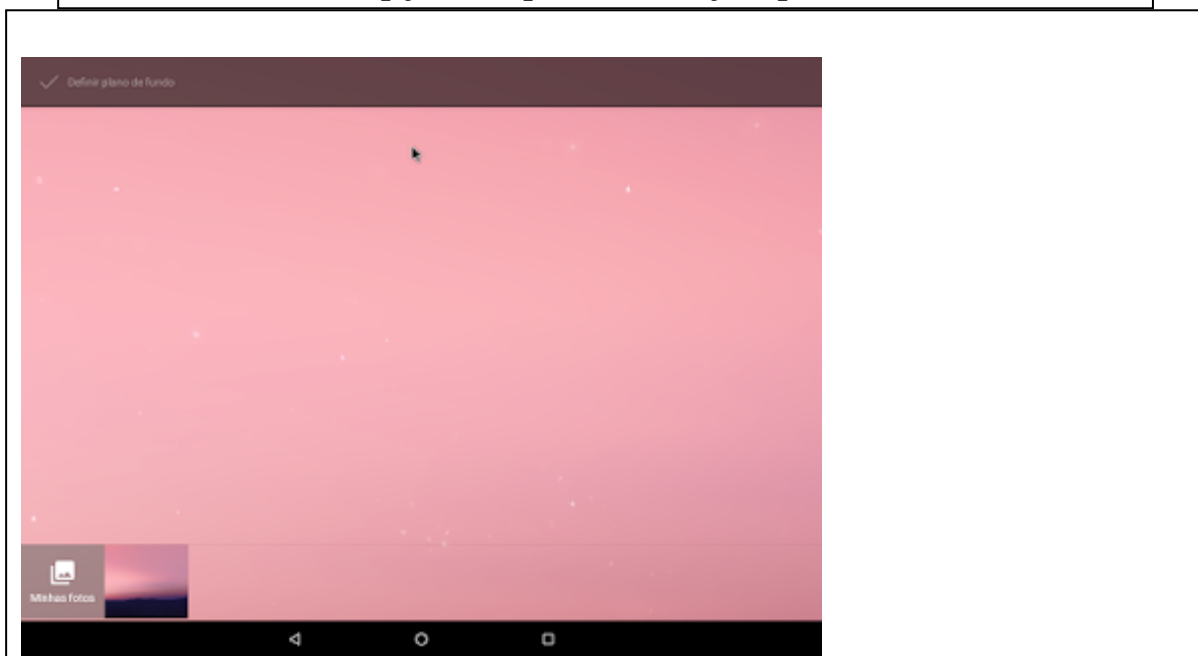
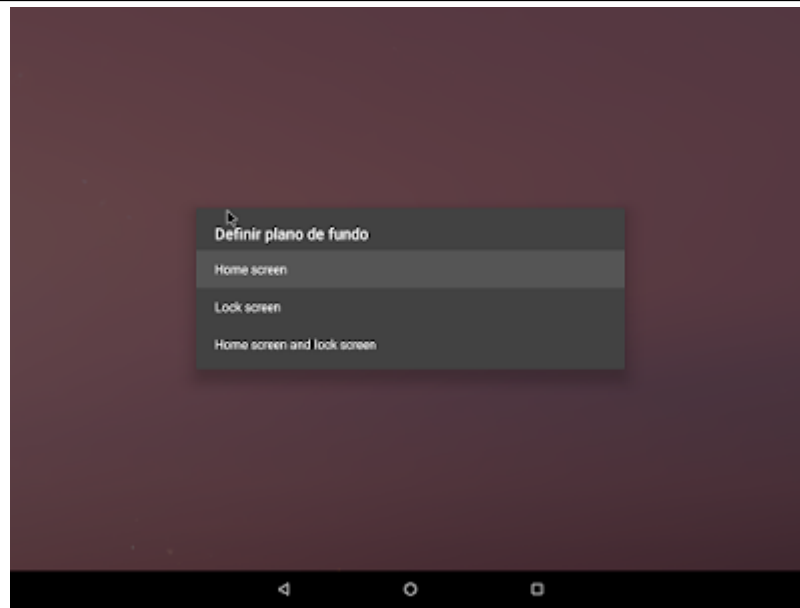
Geralmente, a identificação de conexão é rápida e pode ocorrer sem grandes problemas.

Entretanto, quando não é reconhecido, mostra-se que o Android puro (AOSP) ainda não está preparado para lidar nativamente com conexão Ethernet, focando somente no Wi-Fi (pelo menos, o ícone de conexão cabeada, ao lado da bateria - introduzido no Nougat - já é uma evolução, embora longe da opção dedicada nas configurações do sistema, presente nas compilações ICS do projeto), que é predominante nos disponibilizam que embarcam o SO da Google, o que é uma pena e deveria ser universalizado no futuro, a fim de manter a coesão e de não ser uma tela inútil para quem não utiliza a conexão (contudo, como será detalhado a frente, ela tem, sim, uma utilidade).

Outra tela, ainda inútil no projeto Android-x86, é a busca de atualizações, que serve apenas para perder tempo e que, até aqui, só tem utilidade mesmo nos dispositivos fabricados pela Google, o que não é o caso (dessa forma, como o sistema não suporta, até aqui, o processo de atualização automática, acaba não se beneficiando do Projeto Treble, novidade introduzida nesta versão, que separa as camadas do sistema, limita onde as fabricantes podem mexer e facilitaria a instalação do *update*, com o sistema ainda ligado).

Em seguida, foi adicionada a opção de transferir os arquivos de outros dispositivos ou prosseguir do zero (no Nougat era a segunda tela do assistente, após as boas vindas).

Caso deseje realizar a importação, o assistente perguntará se deseja transferir a partir de outro dispositivo Android, da nuvem ou mesmo de um iPhone (como dá para perceber, dispositivos Windows e distribuições Linux não são opções aqui - quem sabe no futuro).

Caso deseje não copiar/prosseguir do zero, o assistente irá prosseguir a partir desta tela.

Pode ser que você se depare com a tela acima, sem descrição, significando, na prática, a mesma coisa que a imagem anterior.

Esta é a tela "Verificando dados" nativa das versões Nougat e Oreo. Normalmente se prosseguiria para inserção da Conta Google, como ocorre normalmente nas versões estáveis das compilações da versão 7.x do projeto.

Entretanto, como já é uma falha antiga e crônica do projeto, sobretudo nas versões de arquitetura 32 bits (e também já relatado no artigo sobre o Marshmallow, constando oficialmente como problema conhecido), o Google Play Services continua travando e impedindo o prosseguimento da instalação. Pelas minhas análises, o problema não parece ser do *app* em si (que chega a funcionar em alguns momentos, como será detalhado em outros tópicos), mas sim do sistema, além de que continua sendo recorrente, ora resolvido, ora não (uma forma de se esquivar do problema é configurar a conta em uma compilação anterior que não tenha este *bug* - os lançamentos mais recentes do Marshmallow, por exemplo, ou um sistema x64 - e fazer um *upgrade*, de preferência no modo leitura-escrita, como explicado no tópico Instalação).

Embora no Nougat de 32 bits o Play Services não trave, pode ser que, ao tentar se autenticar, "caia" na tela acima, o que originalmente não ocorria, podendo atrapalhar o *upgrade* para uma compilação x86 com o componente relacionado inoperante (além de, na teoria, não haver atenuante, isto é, alternativa para burlar o problema, não está descartado que isto seja apenas temporário ou que possa dar as caras em alguma compilação futura, seja 32 ou 64 bits.

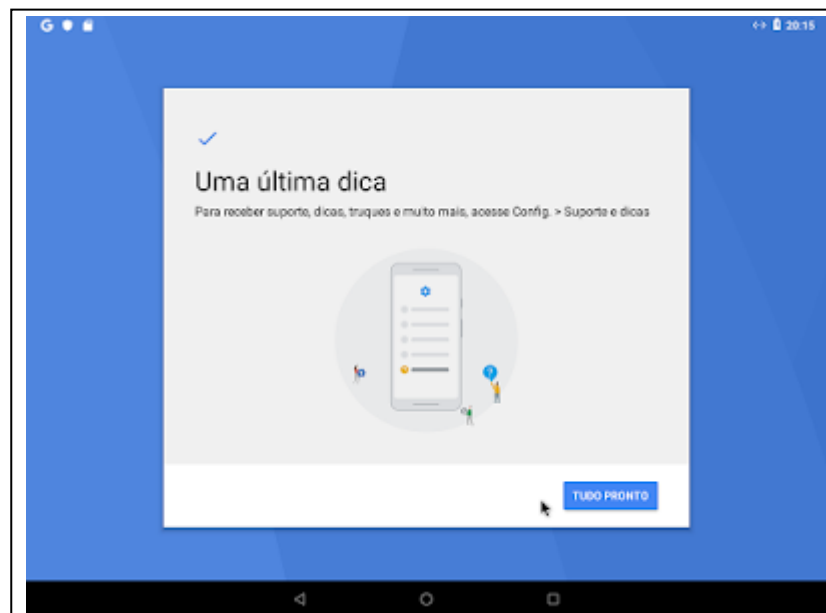
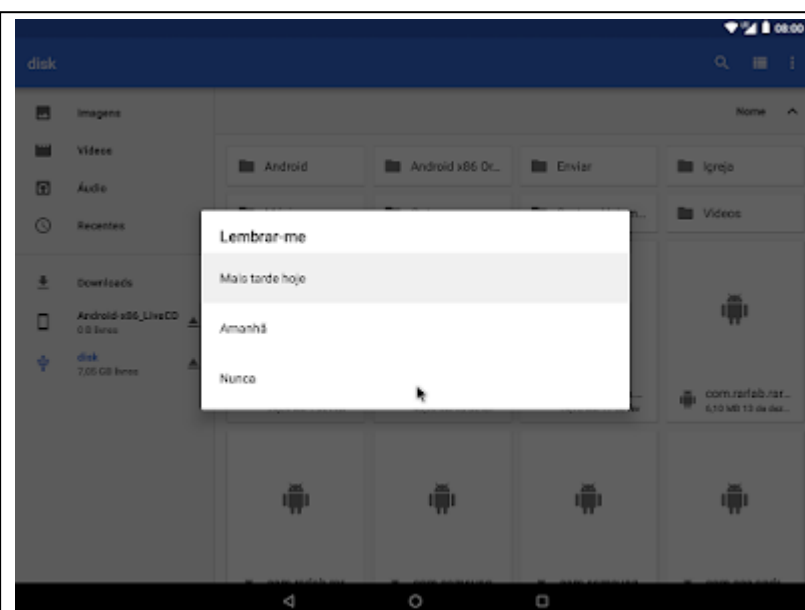
Desta forma, o único jeito de pular este problema e concluir a instalação é pulando a conexão de rede - daí a sua "utilidade" (para tal, pode-se desconectar por alguns instantes - caso a conexão seja Ethernet e/ou esteja instalando numa máquina virtual e reabilitar em seguida).

A seleção de Data e Hora, basicamente, não sofreu alterações.

Agora só é possível inserir o Nome. O campo Sobrenome, presente nas versões Marshmallow e anteriores, foi removido.

Se no Lollipop/Marshmallow era a última, agora a tela chamada Serviços do Google passou a ser a penúltima parte do assistente.

Role até a última opção (ou clique sobre o botão Mais, mostrado na imagem anterior), caso queira revisar as permissões e, no final, clique em Concordo.

Para concluir o assistente, é possível configurar um e-mail a partir do serviço da Google, o popular Gmail ou selecionar outros aplicativos (pode acontecer da segunda opção não ser exibida).

Se antes do Nougat era a penúltima tela do assistente, agora sendo apenas uma opção da última parte, recebeu diversas alterações e um certo destaque, identificando melhor que a tela exibida faz parte do *app* da Google.

Desde o Nougat, também é possível retomar o Assistente, caso necessário, e que será detalhado a seguir.

Primeiro, será solicitada permissão de acesso ao local (repare a referência ao Pixel que permaneceu na compilação).

Em seguida, pode-se configurar uma conexão Wi-Fi de sua preferência, seja manual, seja automático.

O Assistente dá as opções de personalização que são exibidas acima.

Aproveitando, o Android x86 Oreo ainda embarca o mesmo papel de parede solitário das compilações do Nougat como opção (se é que dá para ser colocado dessa forma).

Entretanto, ao definir o plano de fundo, repare que as opções não foram traduzidas para PT-BR (o que é estranho já que, normalmente, faz parte do sistema).

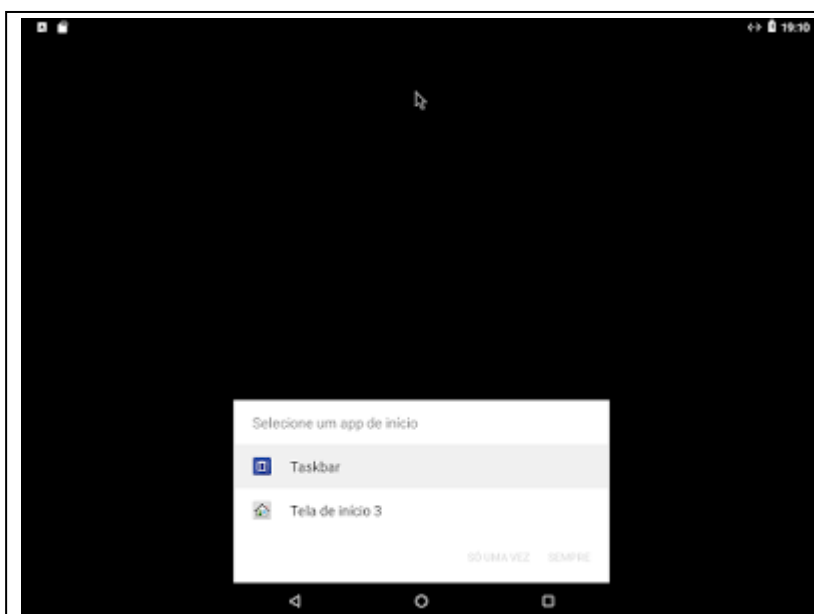
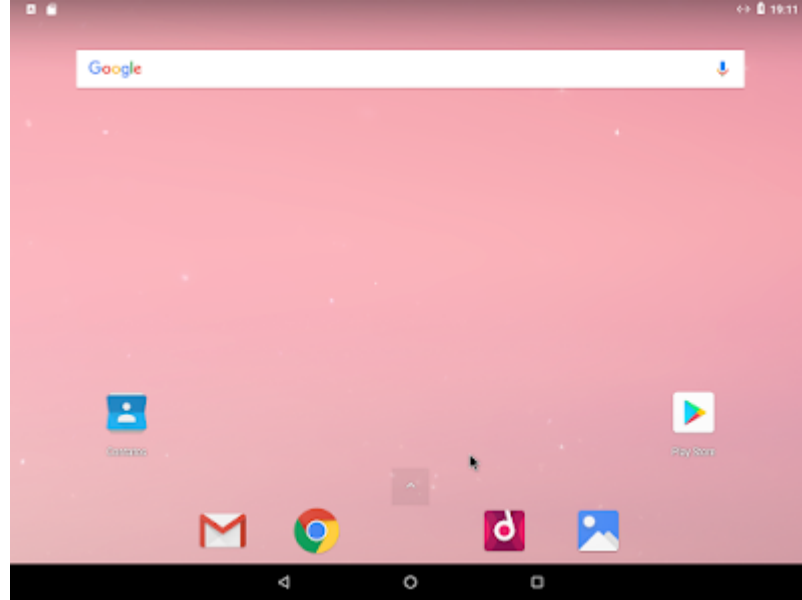
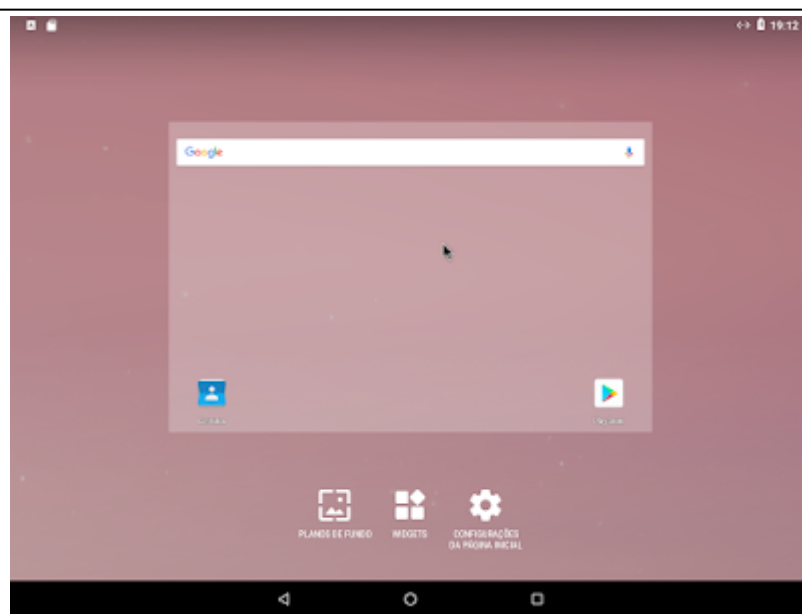
O ajuste do Tamanho da Fonte foi um dos recursos introduzidos no Nougat que facilita o ajuste deste recurso na tela.

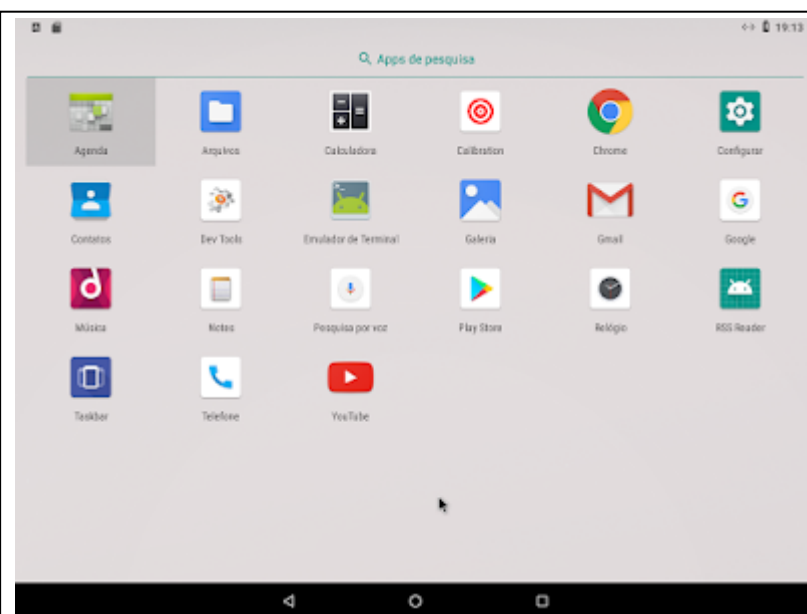
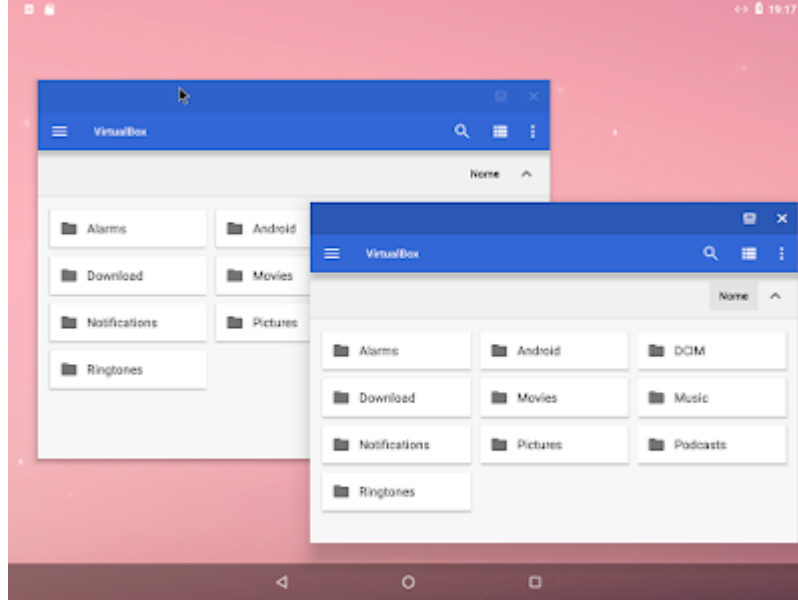
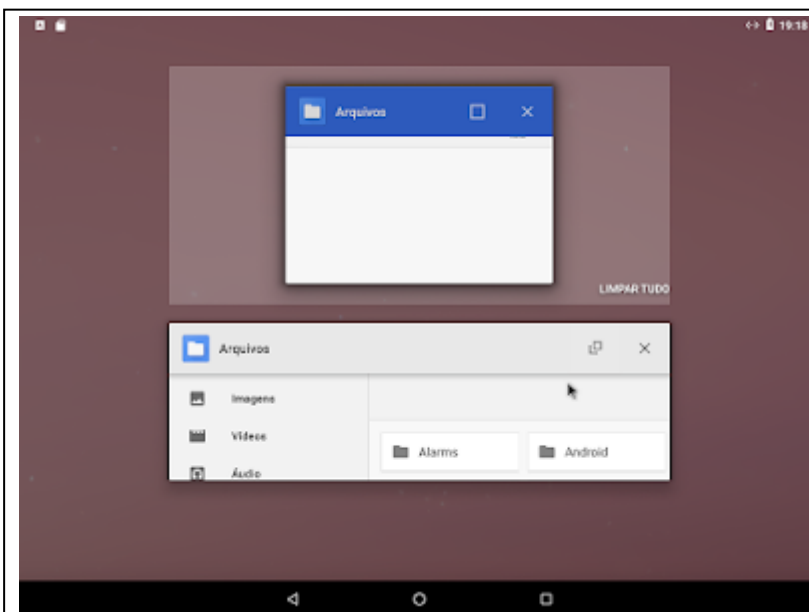
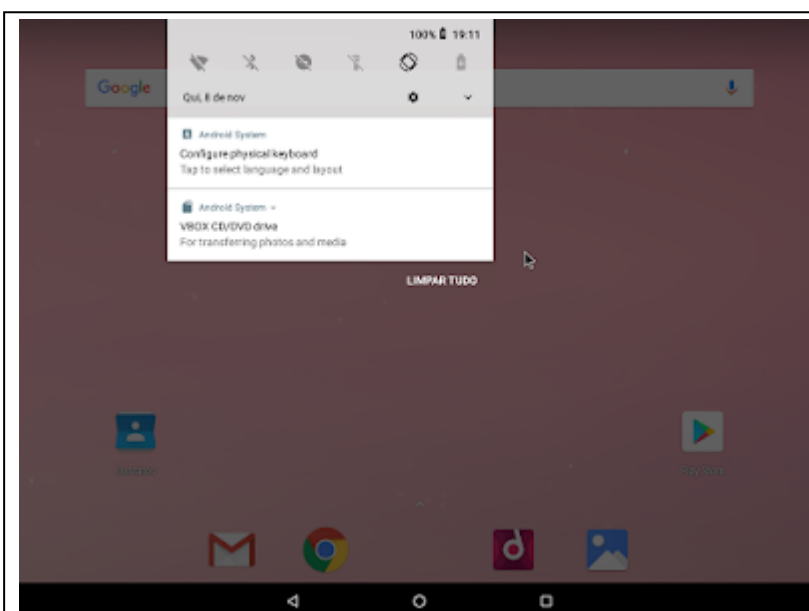
O Tamanho da Exibição, da mesma forma, cuida dos outros elementos da tela.

Esta é a conclusão do assistente.

Vale lembrar que a continuação do Assistente pode ser postergada ou mesmo cancelada.

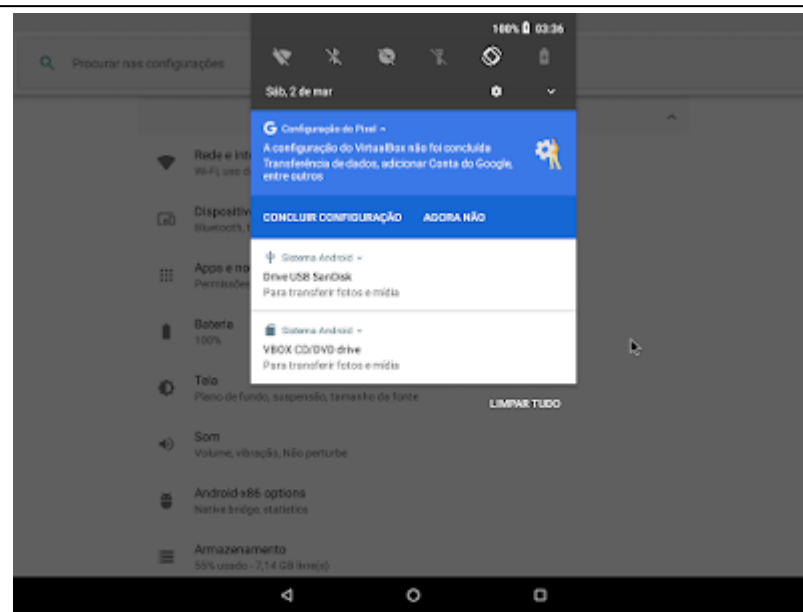
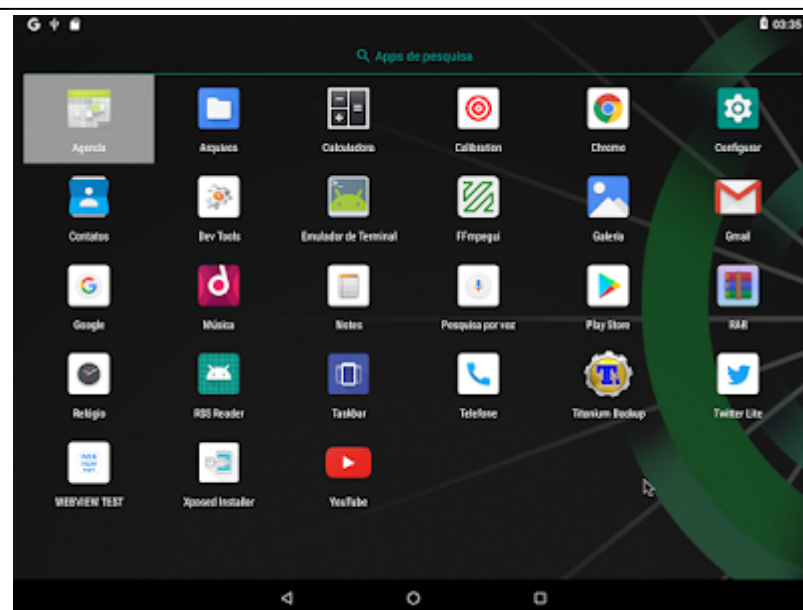
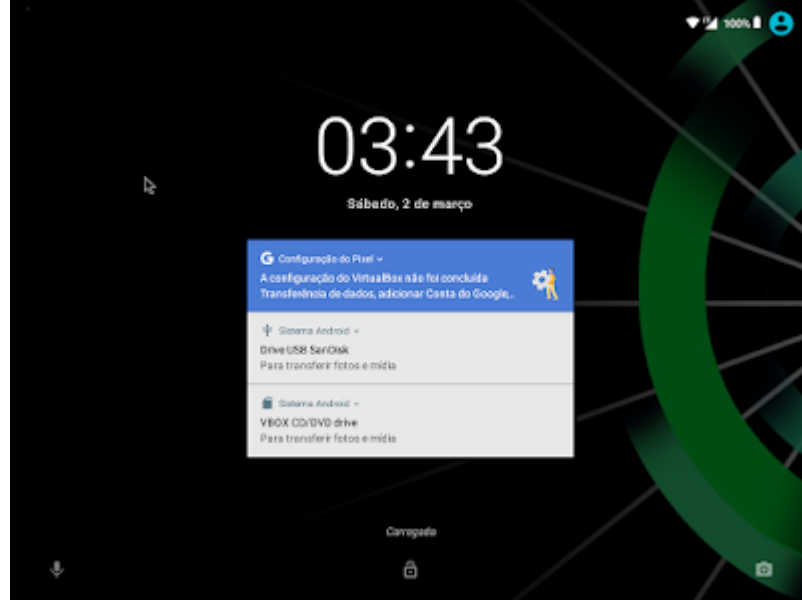
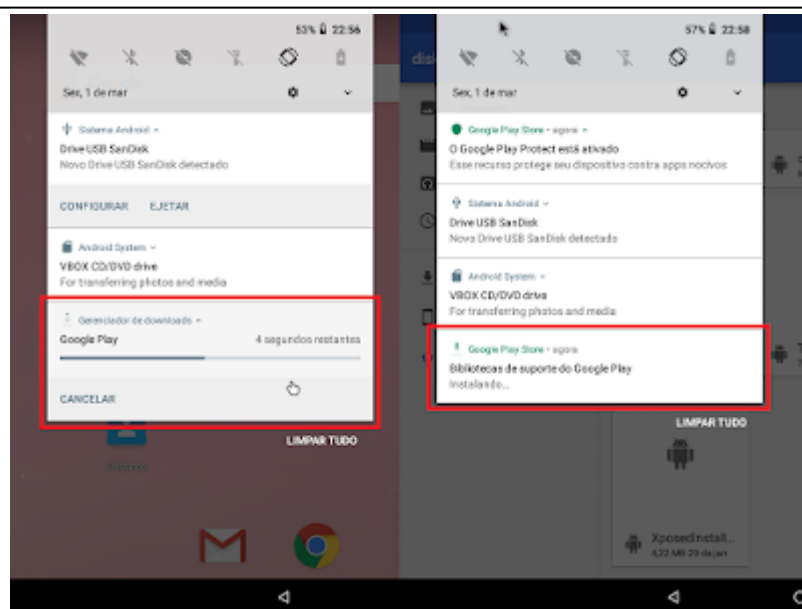
```
[ Minimal BASH-like line editing is supported.  For the first word, TAB
  lists possible command completions.  Anywhere else TAB lists the possible
  completions of a device/filename.  ESC at any time exits. ]

grub edit> kernel /android-7.1-r2/kernel quiet SETUPWIZARD=0 root=/dev/ram0 an>
```

Por fim, aproveitando a opção disponível no modo Live CD, também é possível impedir a execução do Assistente, caso seja optado por reiniciar o sistema após a instalação; para tal, no menu do Grub, **imediatamente na primeira inicialização do sistema instalado fisicamente**, clique sobre o botão 'E' duas vezes seguidas para editar o comando de boot, chegando na tela acima após mover o cursor com o botão '◄' onde, após 'quiet', digita-se SETUPWIZARD=0, então clica-se em ENTER e, depois em 'B'; assim o sistema iniciará em inglês e com algumas configurações predefinidas, como a data e hora no formato e fuso horário correspondente ao idioma citado.

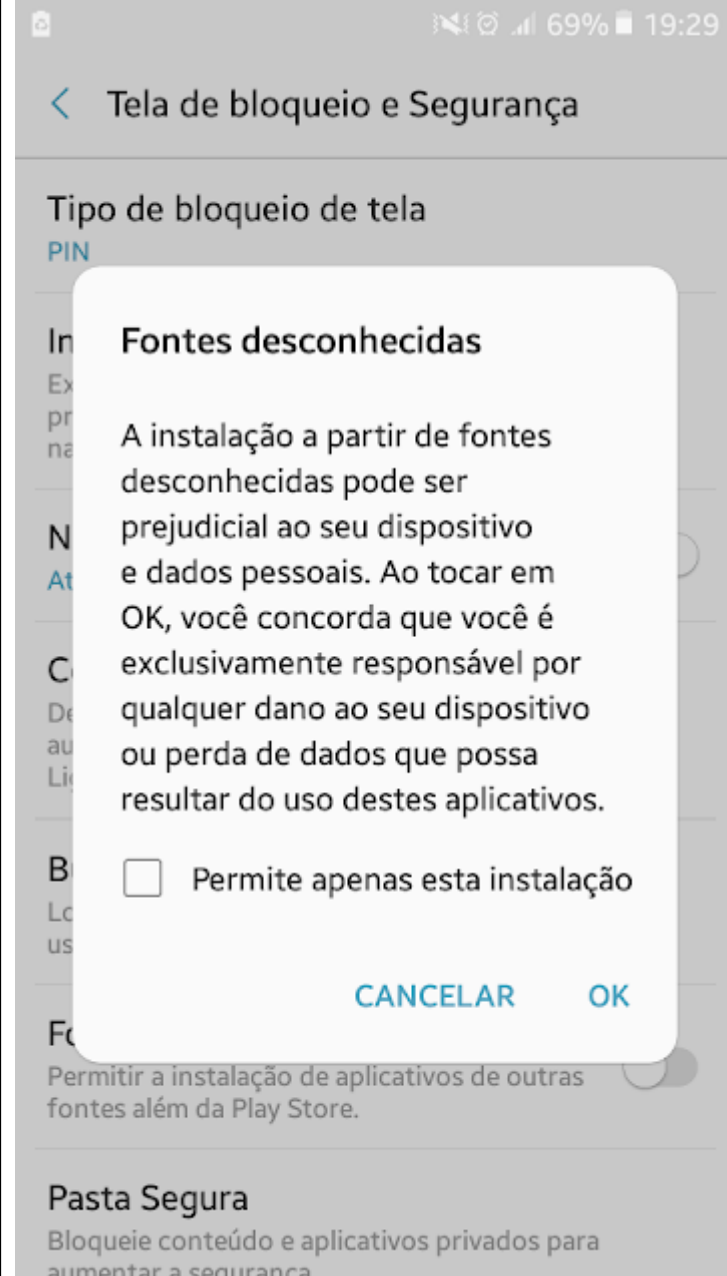
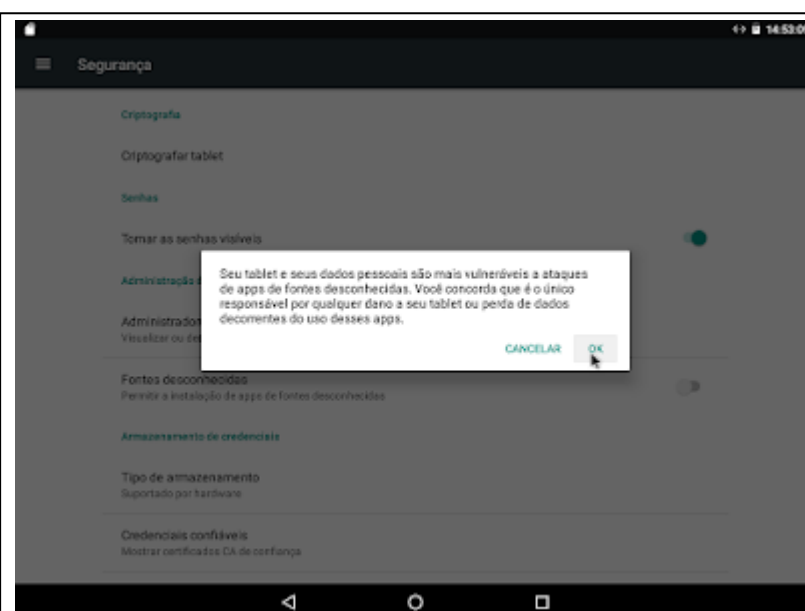
Tela inicial

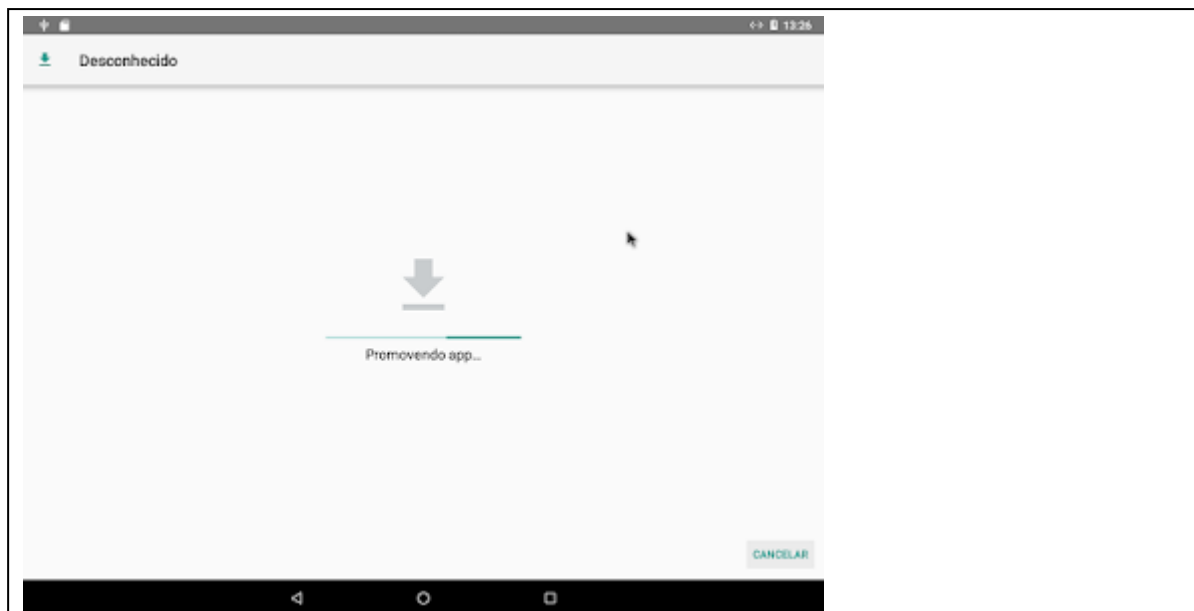
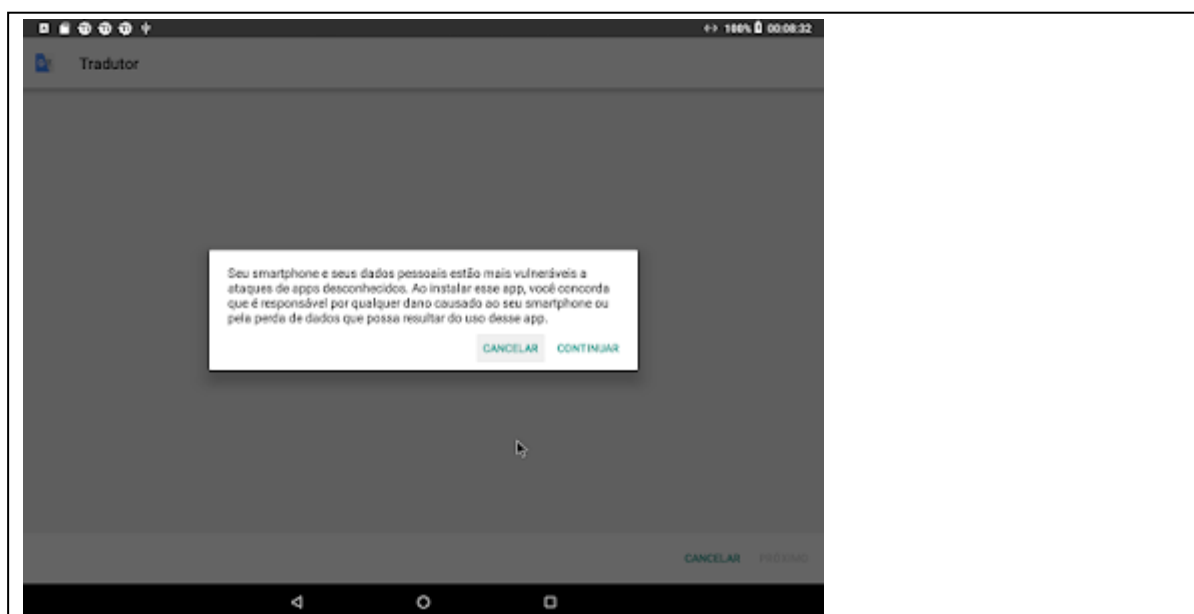
Desde o Nougat, as compilações oficiais do Android x86 passaram a oferecer duas *launchers*:

O lançador nativo, conhecido como Tela de Início 3 e a Taskbar (que será comentada em um tópico próprio). Para o andamento da análise, foi escolhido o lançador padrão do AOSP.

Tela de Início 3 no Android x86 8.1 Oreo R1 (é praticamente o mesmo lançador utilizado no Nougat, não lembrando em nada os padrões utilizados nos dispositivos da linha Pixel).

Ao pressionar a tela, percebe-se que o modo de configuração não mudou desde a antiga Google Now Launcher, com a diferença que o botão "Configurações da Página Principal" é exibido corretamente (a opção é inexistente no Nougat).

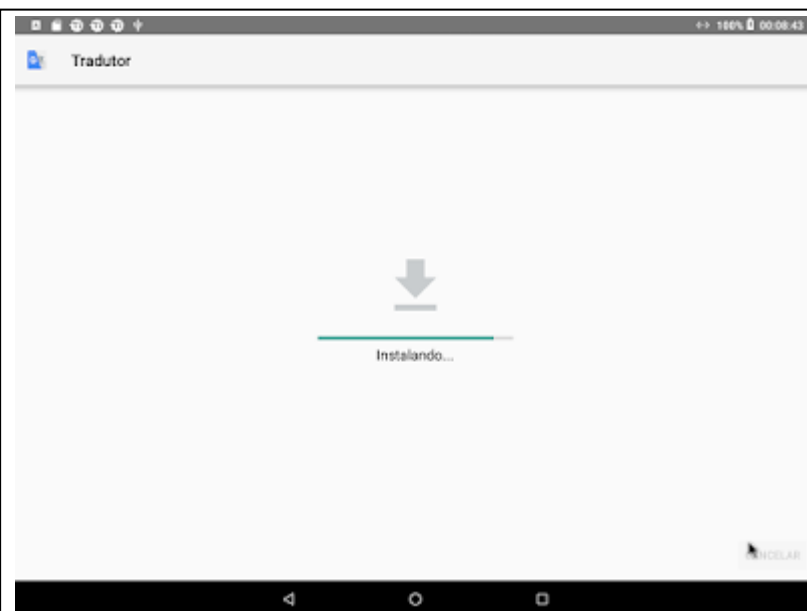
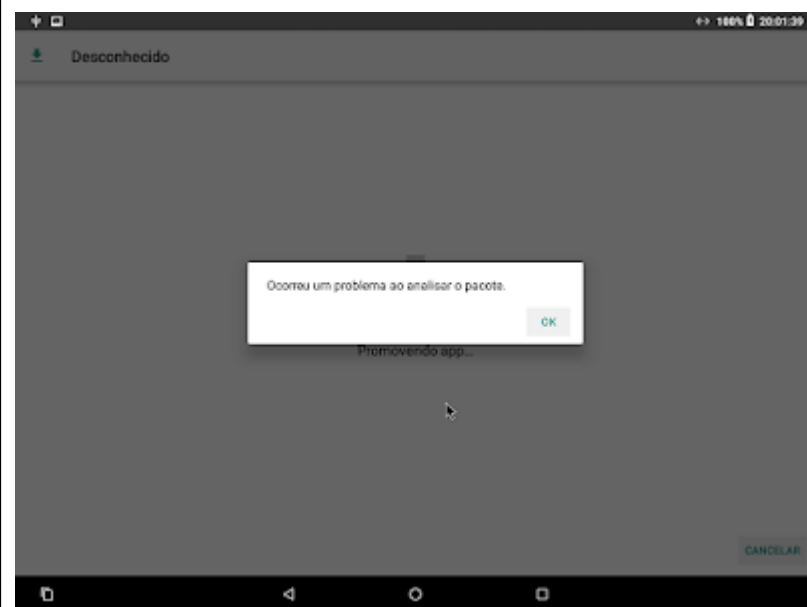
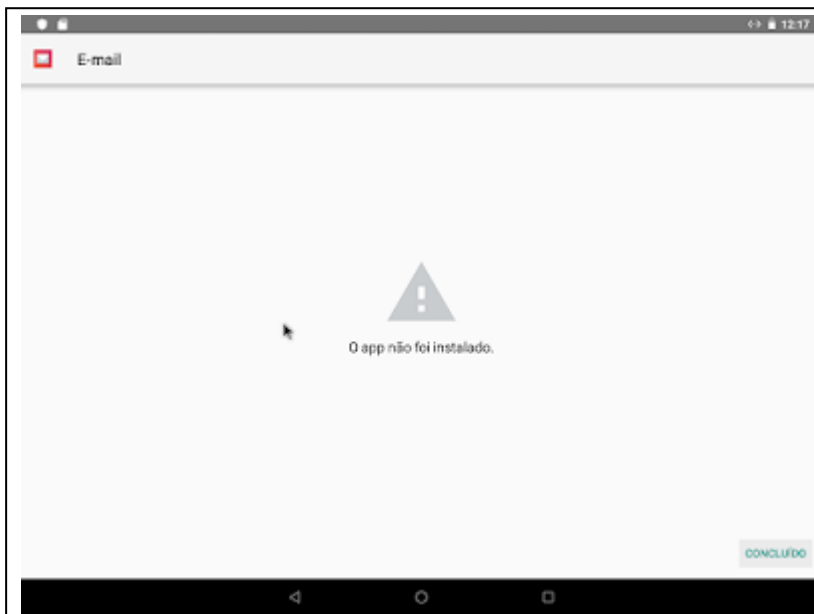
Por padrão, ao acessar o modo de ajustes, repare que, além da opção de fixar aplicativos (movida da Play Store), há o recurso de pontos de notificação (adicionado no Nougat), que identifica o *app* tanto na tela inicial quanto na gaveta se este possui notificações não visualizadas.

A gaveta de *apps*, aliás, não tem tanta diferença em relação ao Nougat, exceto o formato dos ícones (quadrado por padrão - incluindo novos ícones para Configurações e aplicativos sem ícone, como o RSS Reader) e o menu oculto (obtido ao pressionar por alguns instantes sobre o *app*) - recursos advindos do Oreo; e a descrição "Apps de Pesquisa" (no Nougat era "Pesquisar Apps" - o que fazia mais sentido). Vale ressaltar o ícone repaginado do app de Música, mas o que causou-me estranheza os *apps* Calculadora e Relógio estarem com os ícones do KitKat, sendo que, no Nougat, eles eram exibidos corretamente.

Antes embarcado nas Opções de Desenvolvedor como teste, a partir do Nougat o Android permite a execução de múltiplas janelas direto da seção de aplicativos abertos.

Agora, ao acessar os *apps* abertos com o modo habilitado e outro fora do modo, será exibido da forma que é mostrada acima.

A barra de notificações segue o modelo do Nougat, mas recebeu

modificações em relação a este: o relógio e a bateria agora são exibidos no topo (semelhante ao que era no Android ICS); a data e o botão de configurações desceu para abaixo da barra de botões.

Uma das novidades introduzidas no Oreo foi o ajuste automático de cor da barra: se o papel de parede for claro, a barra fica na cor branca (como na imagem anterior); se for escuro, a barra ficará com a mesma cor usada por padrão na versão 7 do Android (repare que as outras telas não são afetadas, mas isto será detalhado num tópico próprio).

E não é só isso: a gaveta de *apps* fica praticamente transparente quando um papel de parede escuro está aplicado.

Esta é a tela de bloqueio padrão, no modo Deslizar, do Android Oreo (em modo tablet).

Desde o Nougat, ao iniciar o sistema pela primeira vez, automaticamente, é baixado a Biblioteca de suporte ao Google Play, que nada mais é que a atualização do Google Play Services, cuja identificação ficou um pouco mais transparente em relação às versões anteriores do SO da Google.

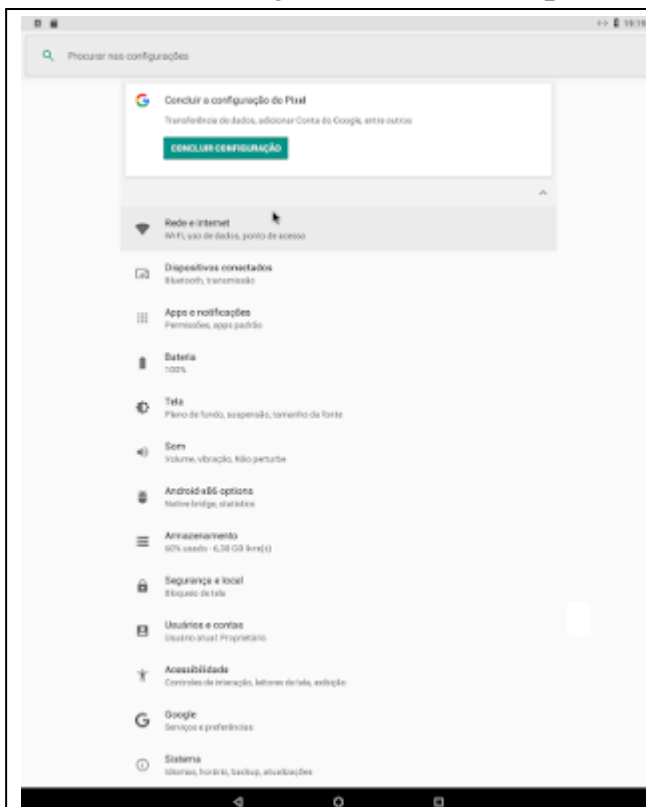
Instalar Apps Fora da Loja

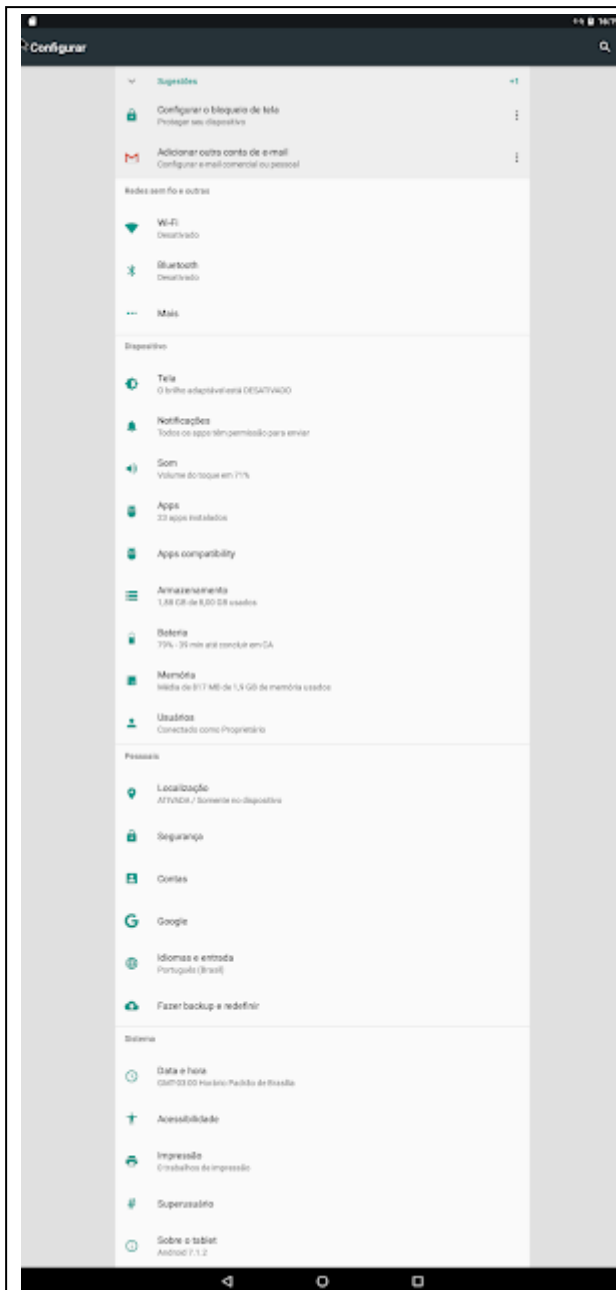
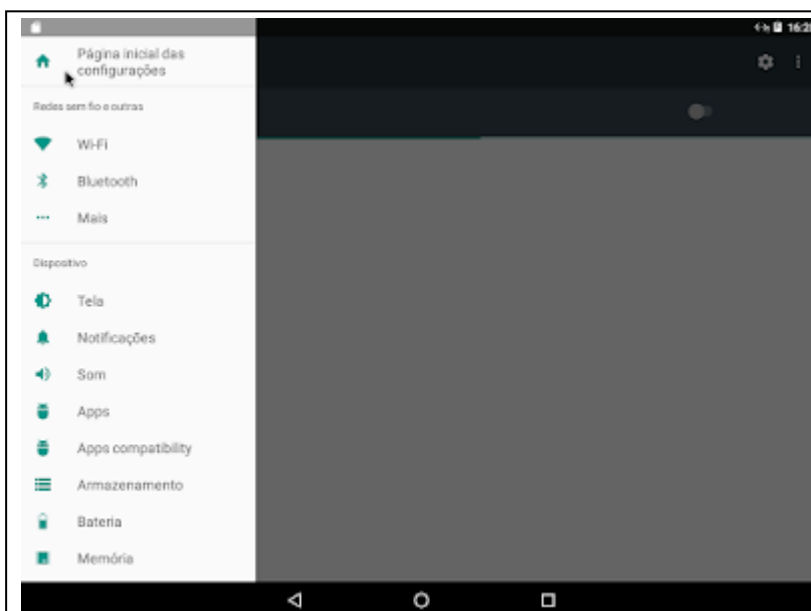
Antes de entrar na tela de configurações, é necessário contextualizar uma das mudanças mais significativas desta edição do Android, que alterou sensivelmente o funcionamento desta função, inalterada desde os primeiros sabores do sistema.

Como se sabe, ao tentar instalar um aplicativo pela primeira vez (ou após reabilitar a opção "Fontes Desconhecidas", em Segurança), o sistema avisa que a instalação está bloqueada e redireciona para ajuste da opção citada (vale ressaltar que, em aparelhos da Samsung, era possível, desde o KitKat, habilitar a opção "Permitir apenas nesta instalação").

Pois bem, a opção vista acima (do Android x86 7.1.2 Nougat R2), presente

desde os primórdios do sistema, já não existe mais no Oreo.

O processo, que foi remodelado no Nougat, agora, propositalmente, demora mais para processar o *app*.

Dessa forma, toda vez ao instalar um *app* fora da Play Store (e outras lojas de aplicativos oficiais), o usuário é notificado dos riscos e se deseja prosseguir.

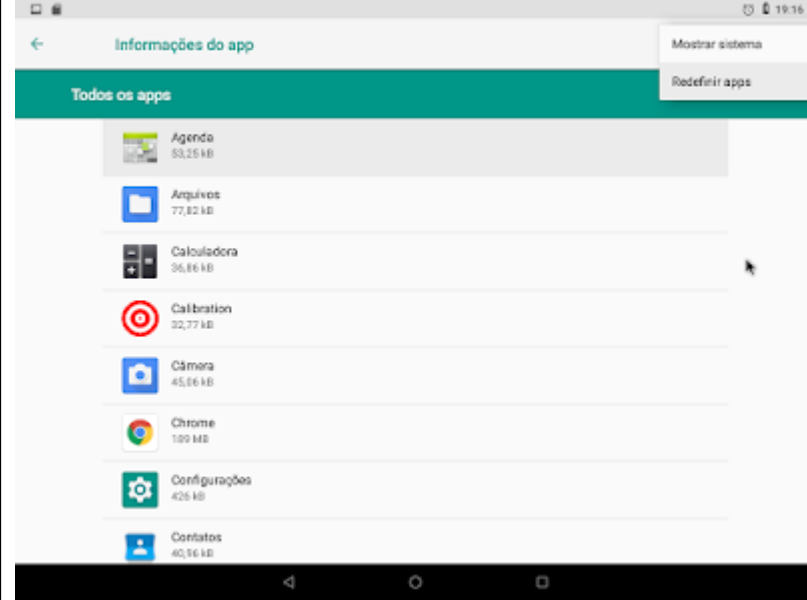
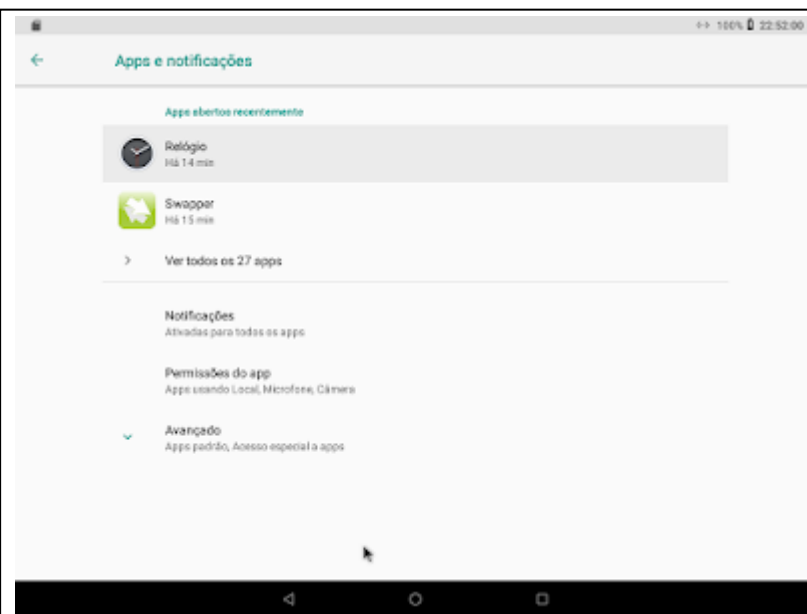
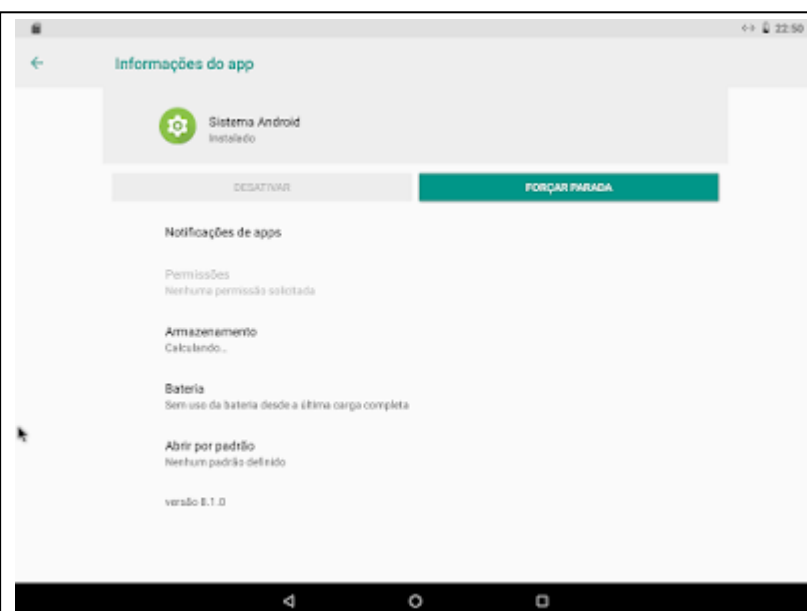
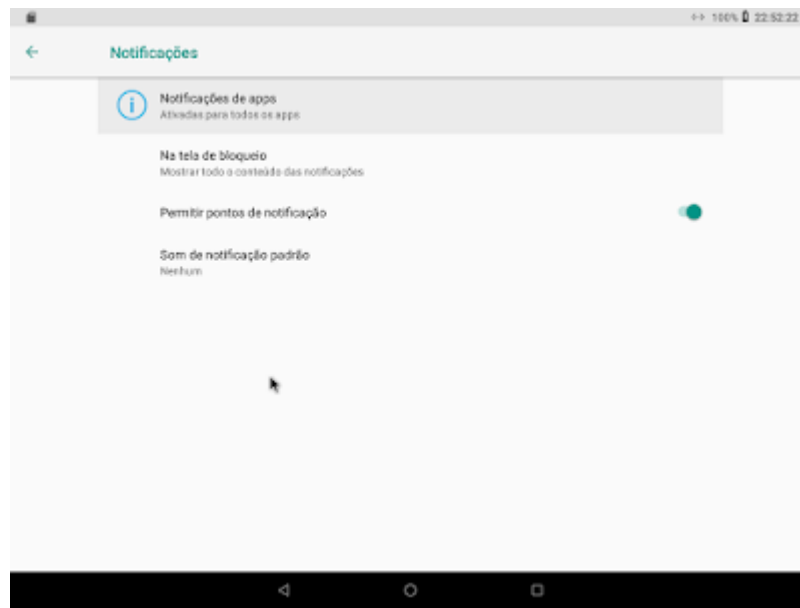
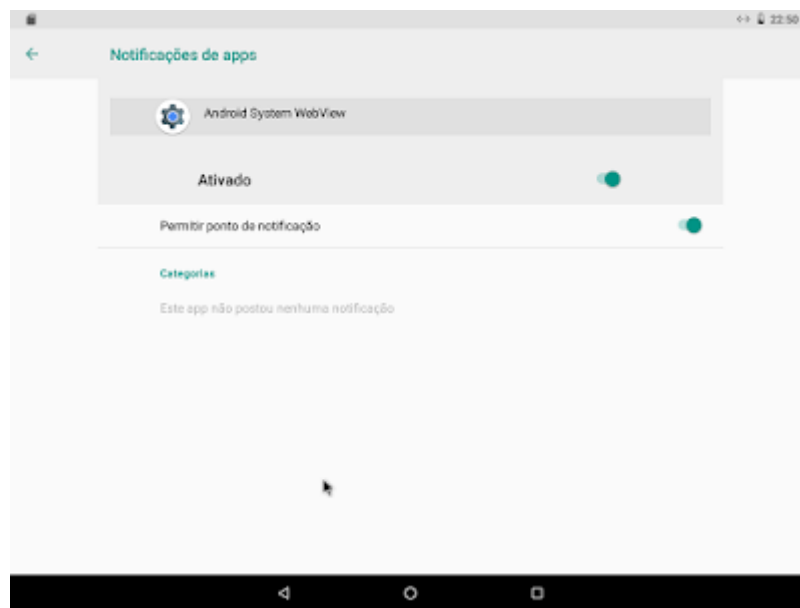
A tela, praticamente idêntica à versão anterior, segue mais ou menos o mesmo padrão conhecido desde sempre.

Contudo, ao instalar, a barra de progresso teve o comportamento alterado, funcionando mais ou menos como no Windows onde, ao chegar na posição mostrada na imagem acima, o botão Cancelar (sim, uma das novidades desta versão) é desabilitado até a conclusão.

Além disso, um dos problemas é a falta de confiabilidade do instalador, que recusa a instalação de determinados APKs que deveriam ser reconhecidos, sem motivo aparente (e, em meus testes, estes mesmos APKs instalam ou atualizam com sucesso no modo texto, utilizando o comando *pm install*).

Também remodelado no Nougat, a imagem acima mostra quando a instalação do *app* falha.

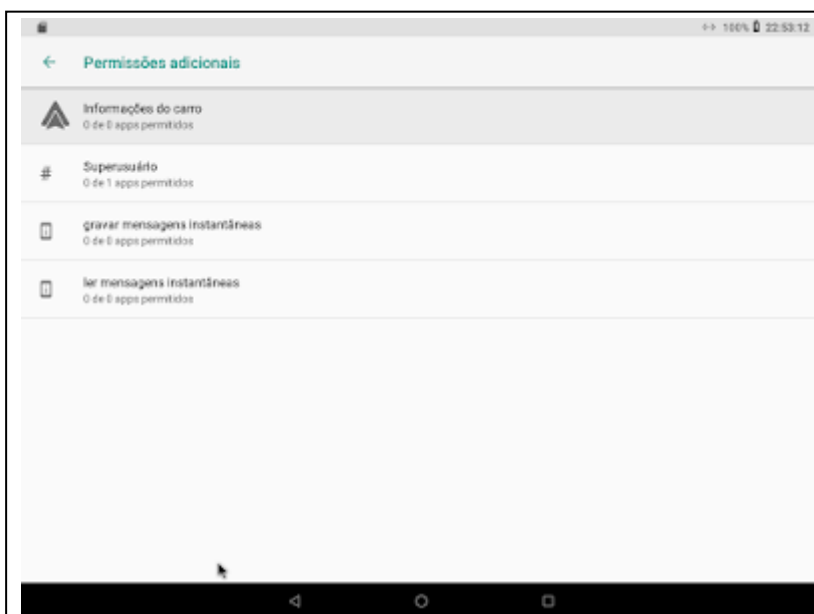
Configurações

Sofreu modificações bastante expressivas em relação ao Nougat.

Além de ter uma barra de pesquisa habilitada por padrão (que, particularmente, não é harmônica em relação ao resto da tela), as opções ficaram enxutas, mais objetivas e descritivas (a versão anterior só fazia

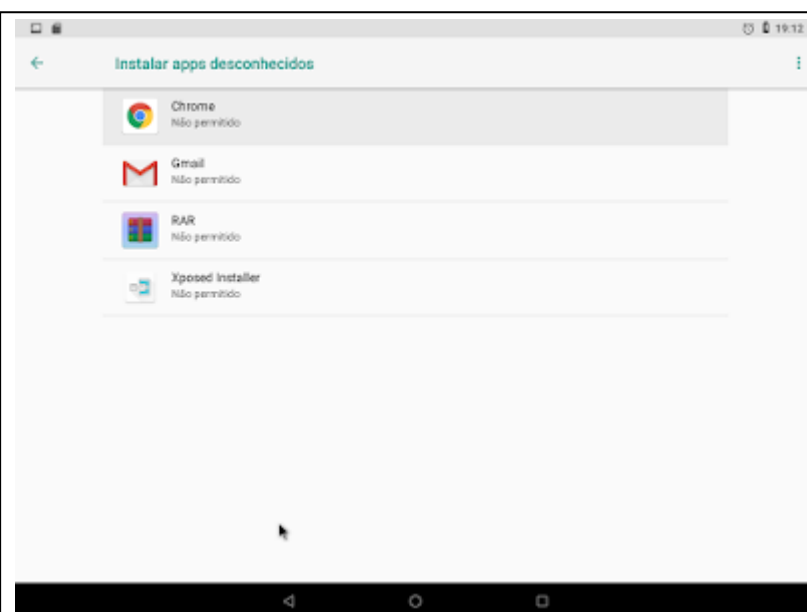
isso em algumas opções) e o espaço de sugestões (também introduzido no Nougat) foi aprimorado.

Tela de configurações no Android x86 7.1.2 R1 (repare que a tela é mais próxima do Marshmallow que do Oreo, claramente mais extensa que o último).

Em compensação, o menu hambúrguer, disponível ao abrir qualquer opção da tela principal, não está presente no Oreo (pelo menos, não no formato de tela padrão).

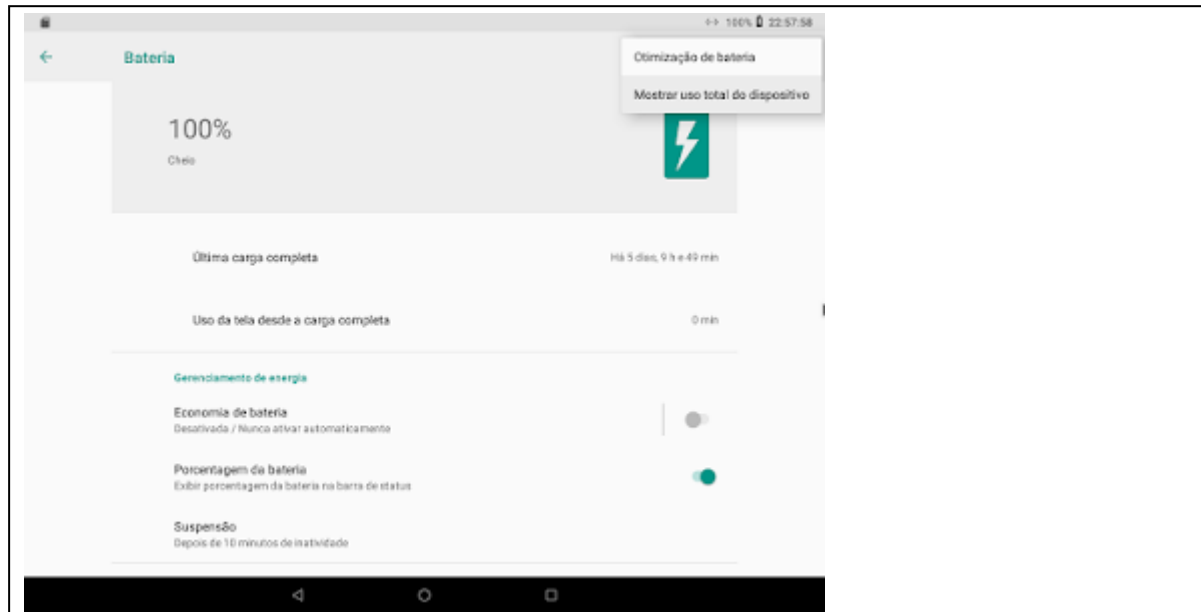
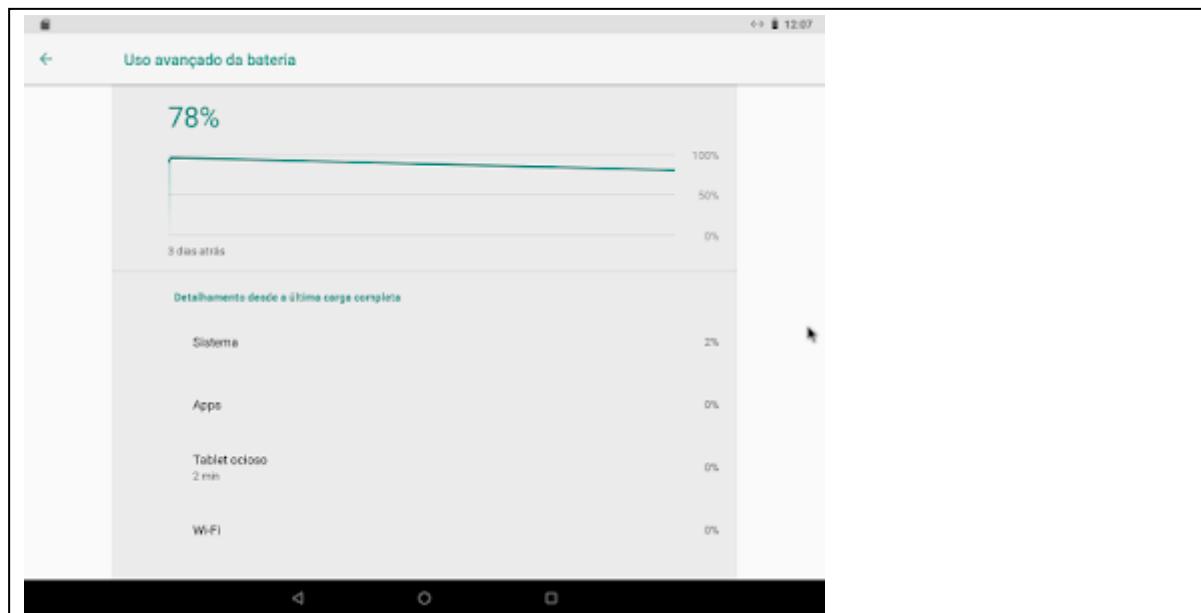
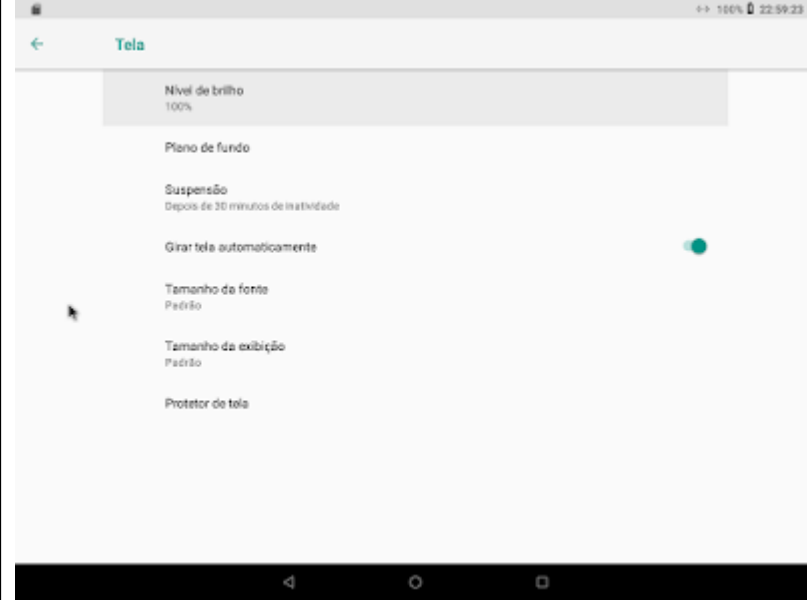
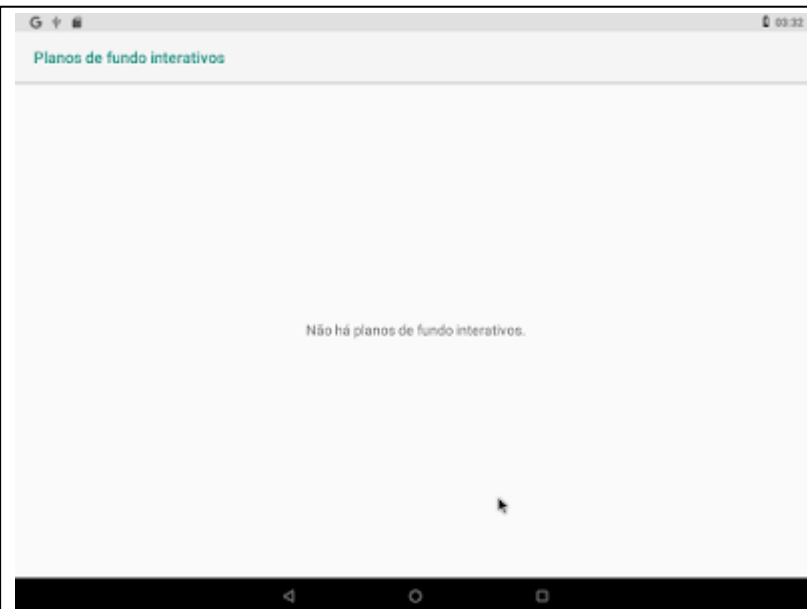
Mesmo citado na opção avançada, a configuração do Bluetooth foi movida desta tela.

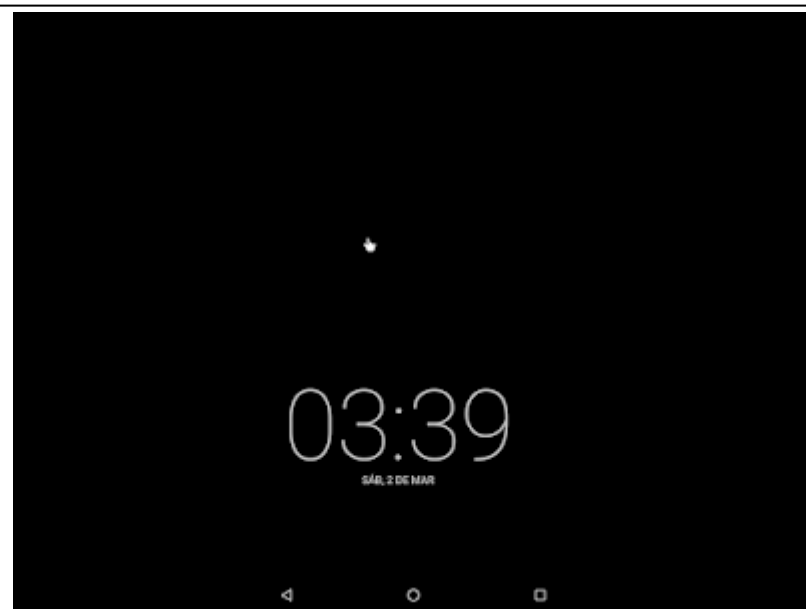
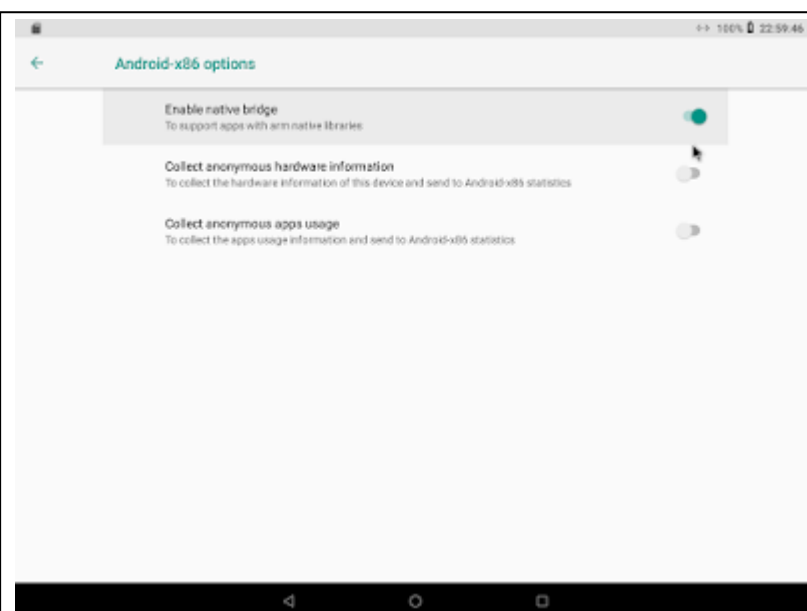
Tela Dispositivos Conectados (repare como o Bluetooth foi separado dos outros tipos de rede para se integrar com outros métodos de conexão entre dispositivos, como impressão - que era seção principal em versões mais antigas do Android - e USB - que só podia ser configurado nas Opções de Desenvolvedor, apesar da escolha do modo ainda permanecer por lá).

A Tela Informações do app, basicamente, segue o mesmo padrão desde o Marshmallow, mas ficou um pouco mais colorida, além de ter perdido o botão da engrenagem.

Tela Apps e Notificações (sofreu uma forte repaginada, ganhando uma tela resumo, com os *apps* abertos recentemente e principais opções, que eram escondidas no botão da engrenagem).

A tela Informações do *app*, em si, não mudou muito desde o Marshmallow (as únicas diferenças são os botões maiores e coloridos e, principalmente, a versão, que foi para o final da tela).

A opção Notificações reuniu as principais configurações, que antes eram espalhadas pelo sistema.

A configuração de notificação individual, que sofreu aprimoramentos a partir do Nougat (permitindo a configuração direto na barra de notificação, por exemplo), recebeu mais mudanças no Oreo, tornando-se mais objetivo, com possibilidade de visualizar as notificações mais recentes e, caso o aplicativo suporte, escolher quais itens podem ser notificados (por exemplo, no Chrome, é possível desabilitar somente a notificação de Downloads ao invés de afetar toda a funcionalidade).

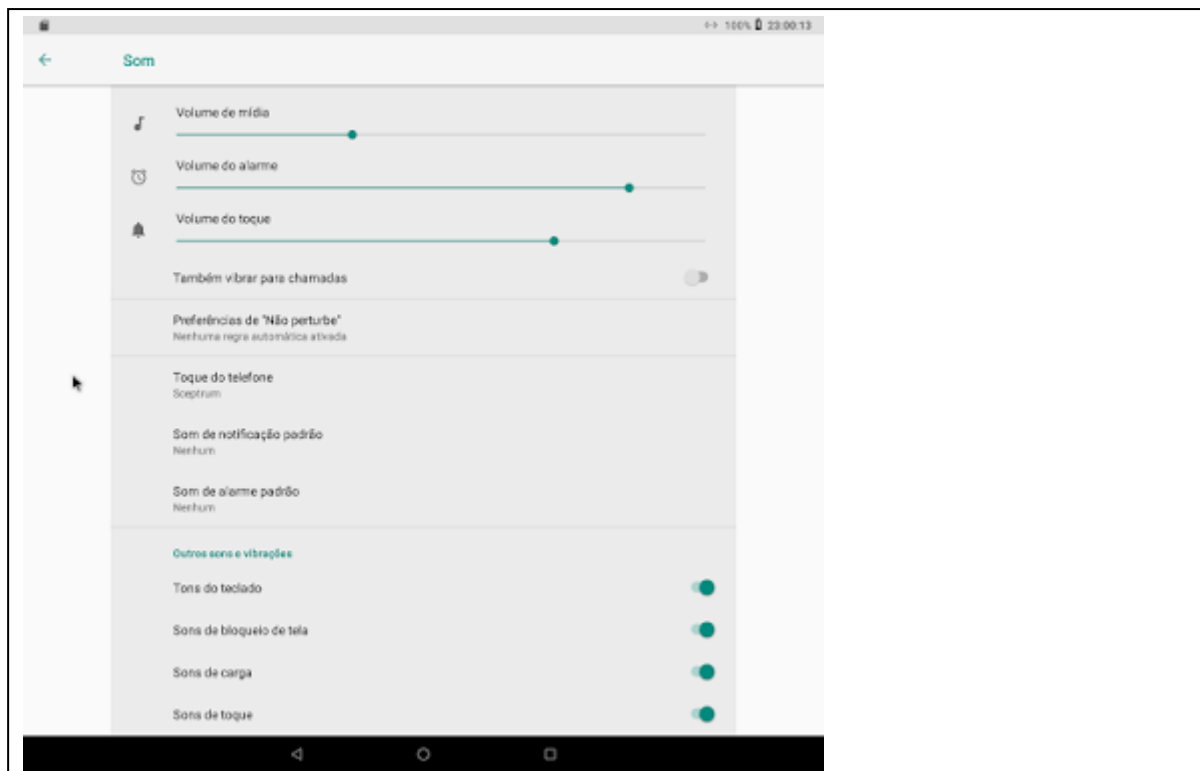
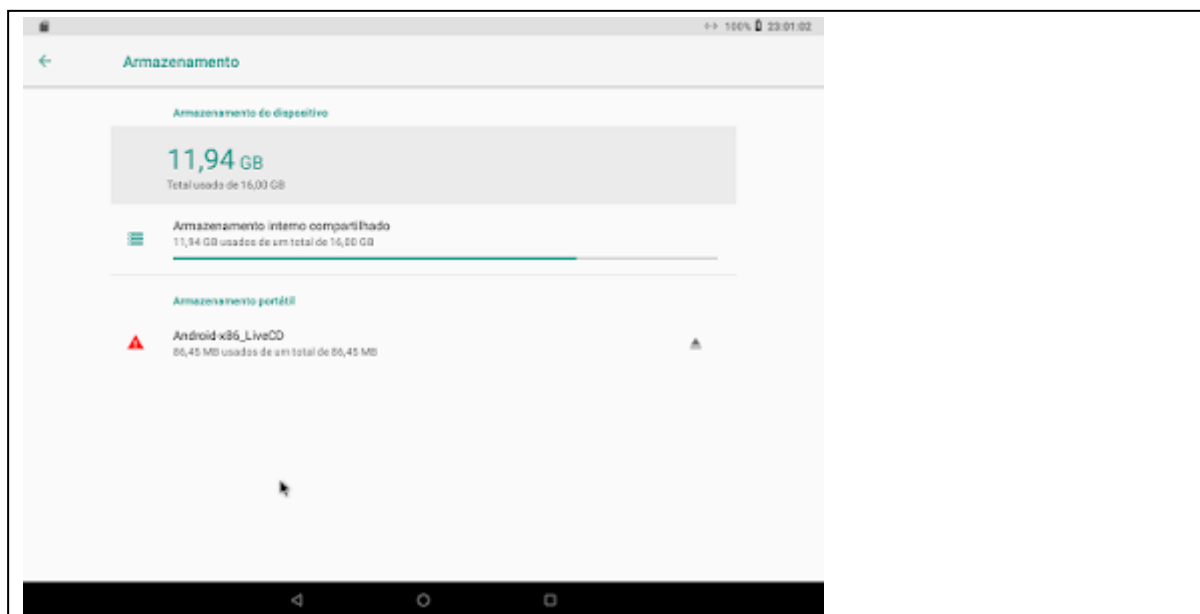
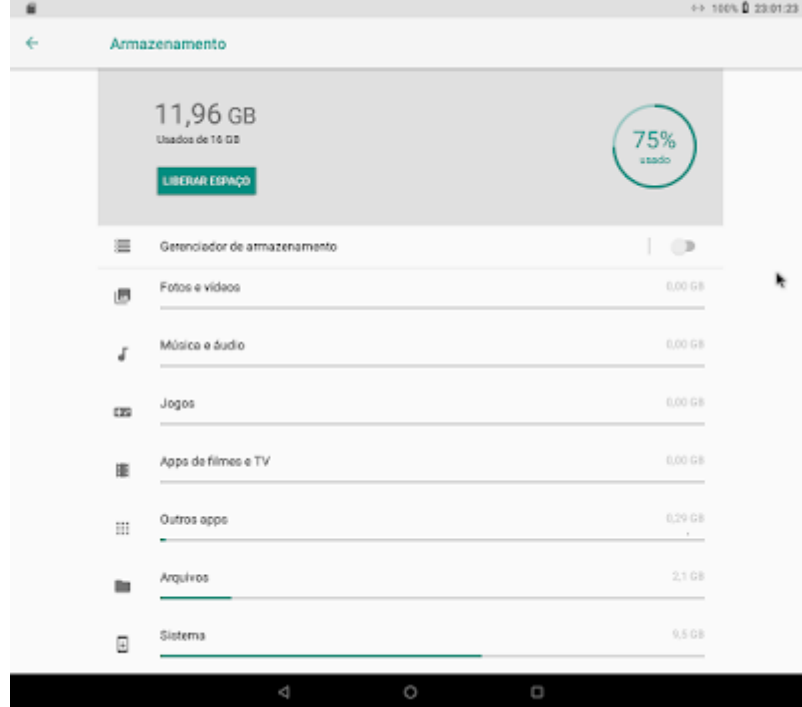
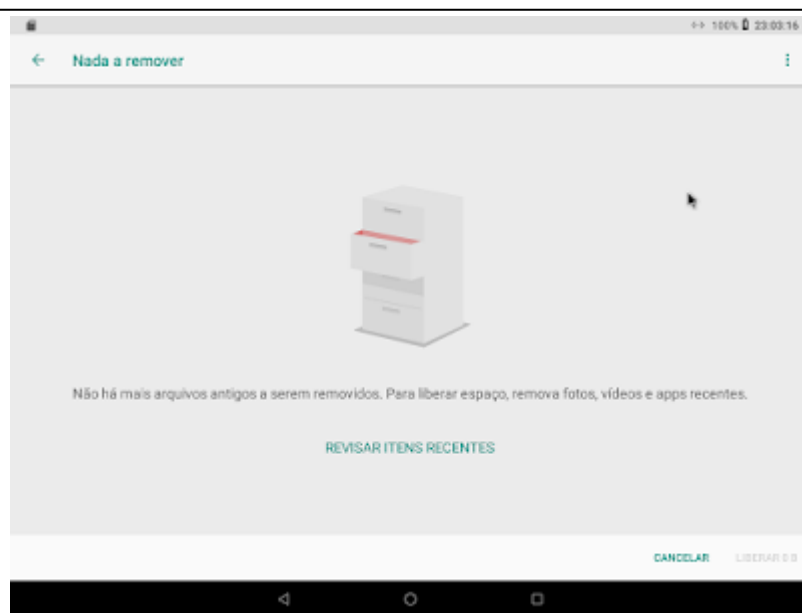
A Tela Permissões do *app* praticamente não mudou desde o Marshmallow.

Aproveitando para ressaltar uma característica de ROMs com o modo *root* habilitado, o Android passou a suportar, desde o Nougat, a seleção nativa de *apps* que podem usufruir do acesso privilegiado (mas, por ser simplório e, no caso do Android x86, não ser tão preciso, seu uso não será muito efetivo).

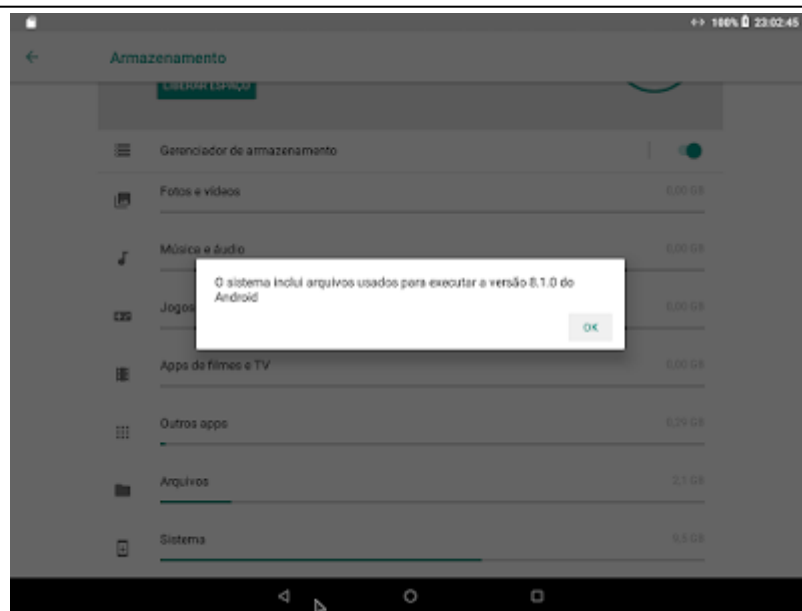
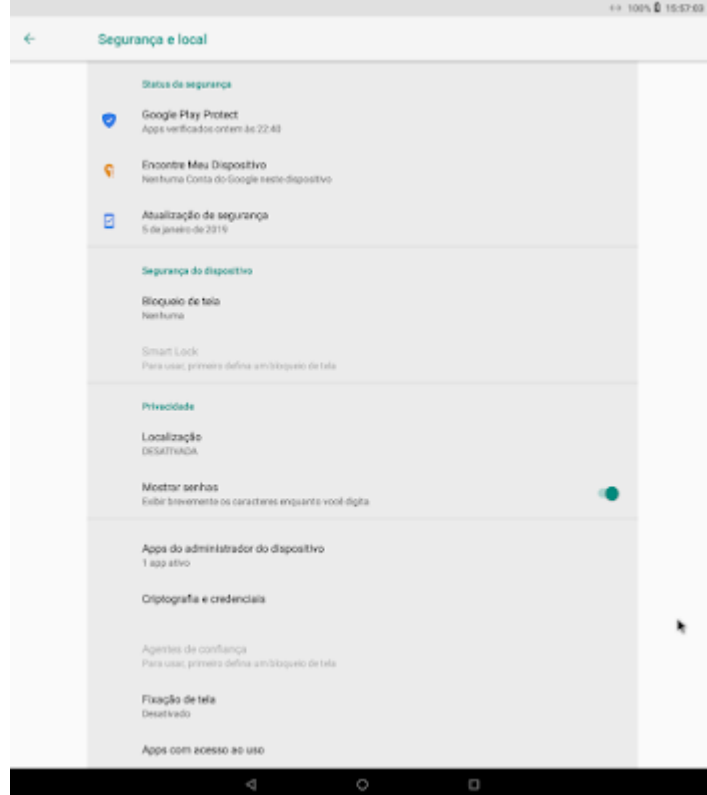
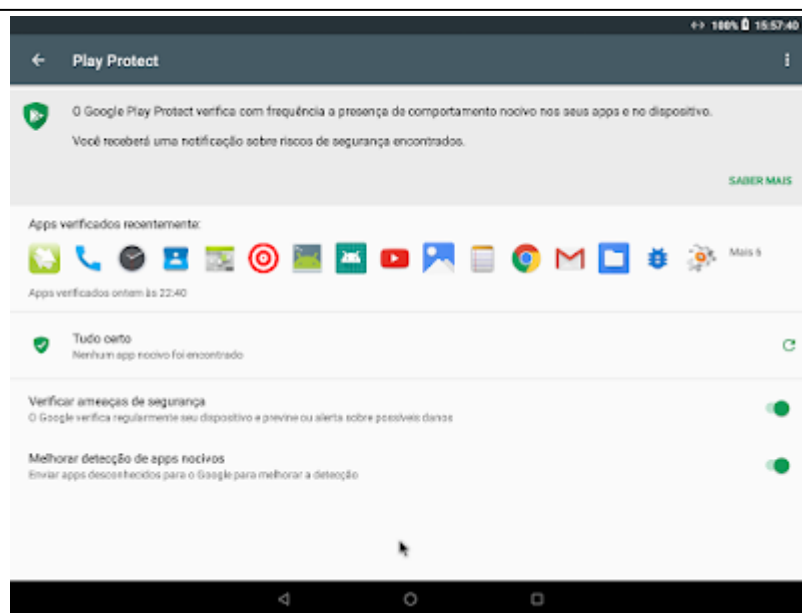
No caso, listou apenas o Emulador de Terminal, mesmo com o Titanium Backup instalado.

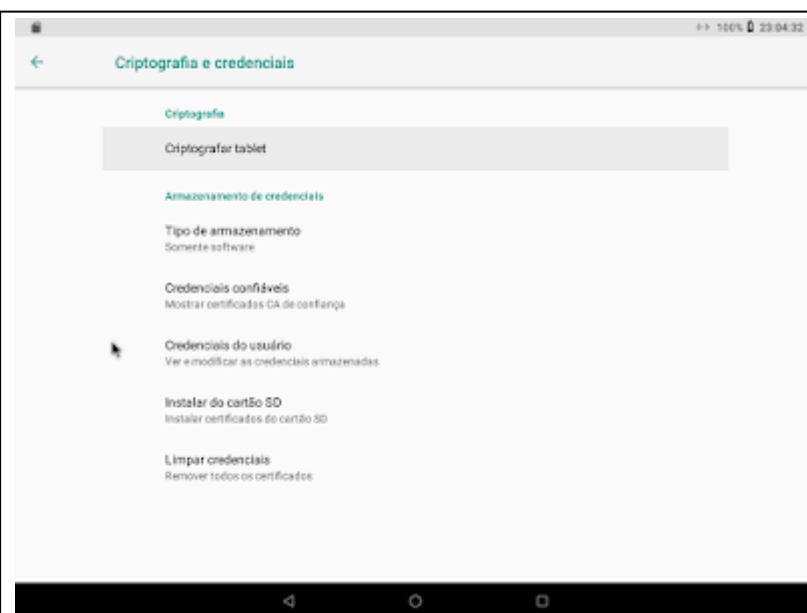
A Tela Acesso especial a *apps* (no Nougat, somente Acesso especial) foi aprimorada e ganhou novas opções, tais como *Picture-in-Picture* e Instalar aplicativos desconhecidos.

Parte da mudança do processo, a opção mostrada na imagem acima busca

permitir certas aplicações, que não sejam, necessariamente, lojas de aplicativos, a instalar *apps* sem confirmação. No caso, dentre outros, o Gmail foi listado (o que é um tanto curioso, já que o aplicativo de e-mail da Google, em tese, não tem esta função).

A Tela Bateria também sofreu bastantes modificações em relação ao Nougat, onde foram adicionadas as opções Porcentagem de Bateria (até então escondida no Sintonizador System UI) e Suspensão (duplicada da opção Tela).

Ao clicar sobre o banner que informa o valor da bateria, tem-se acesso a esta tela que, basicamente, é um aprimoramento da tela Bateria presente do Lollipop até o Nougat, substituindo a tela clássica, presente desde as primeiras versões do Android, que tinha a função, nas versões citadas, de mostrar o uso avançado.

Já em Tela não houve muitas mudanças (Modo de espera virou Suspensão - mudança infeliz, o que ficou supérfluo, como pode ser visto em Bateria); perdeu a opção de Controle de Rotação da tela (agora presente só na barra de botões nas Notificações) e Transmitir (que foi movido para Dispositivos conectados).

Desde o Nougat o Android x86 não traz mais nenhum plano de fundo interativo.

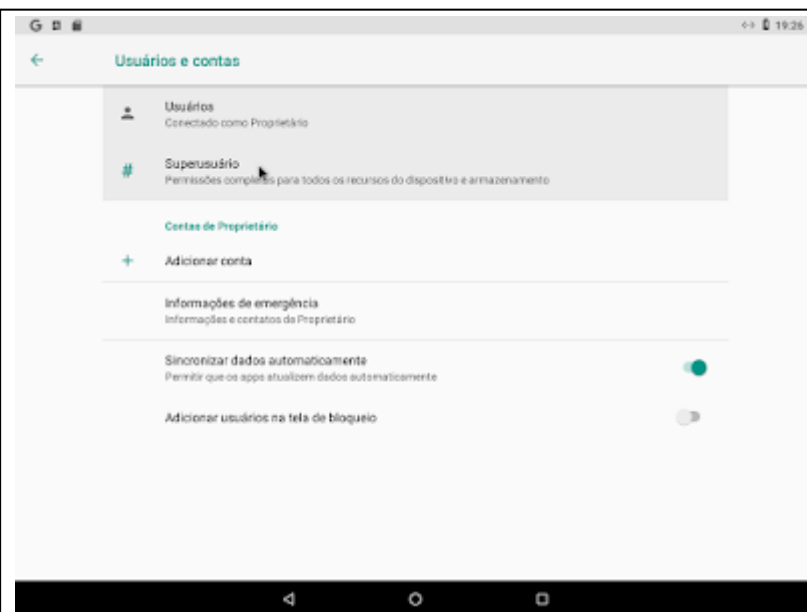
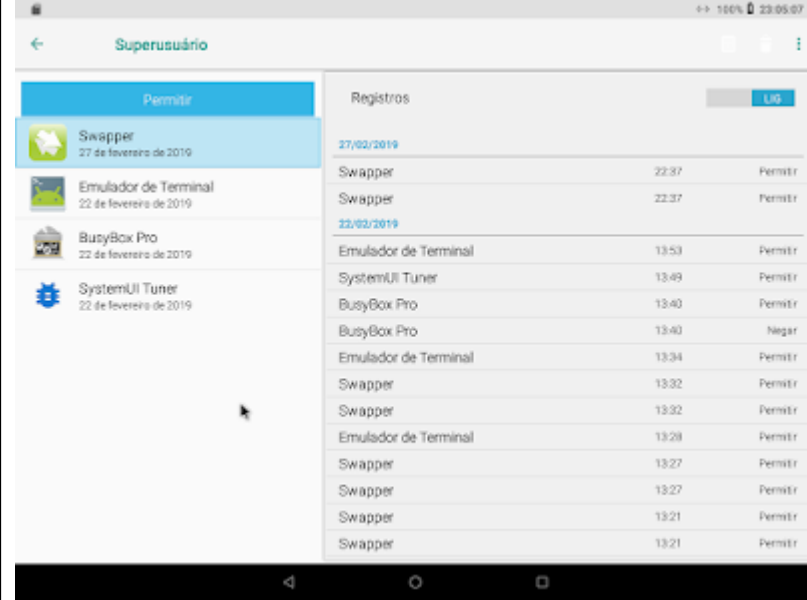
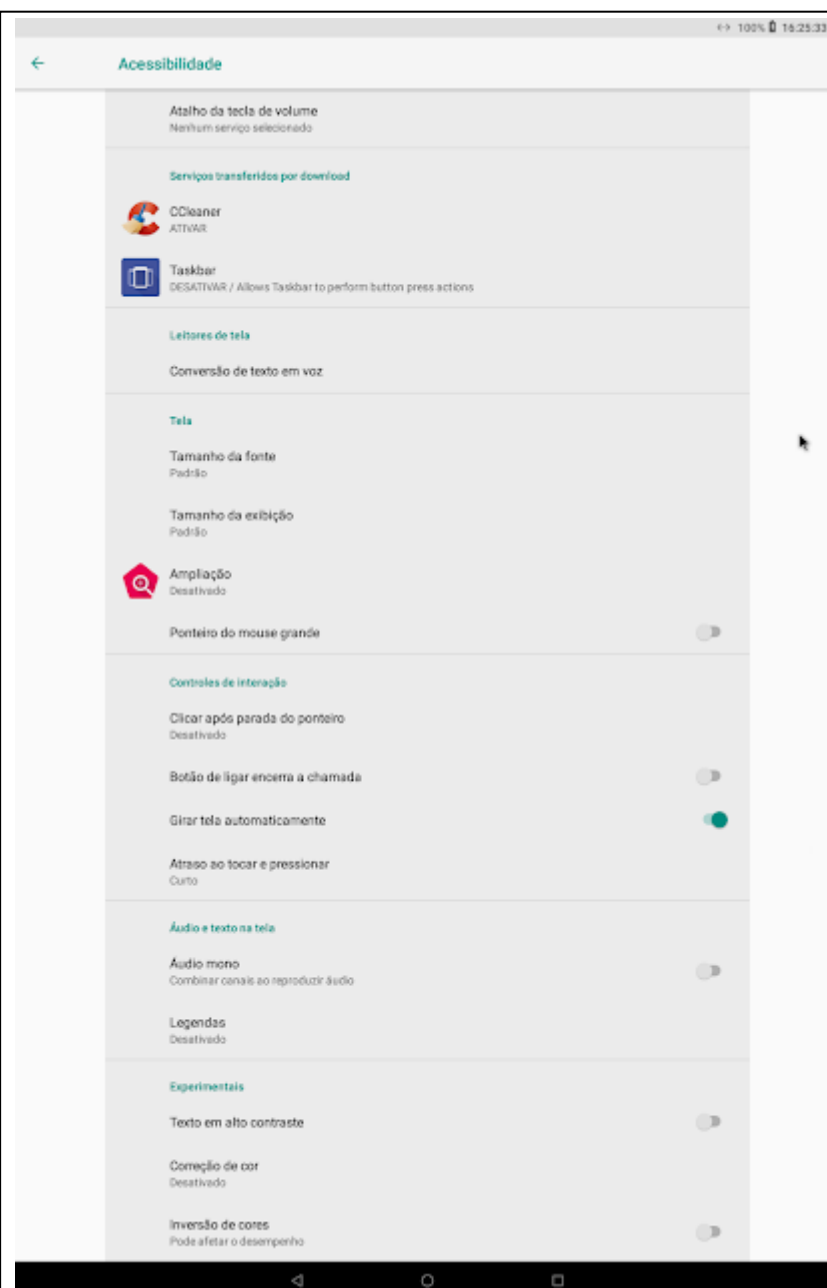
O Android Oreo ainda traz o recurso, que um dia foi chamado de Daydream.

Curiosamente, o modo relógio ainda traz o mesmo modelo usado no KitKat.

A Tela Android-x86 Options (antes Apps compability) recebeu novas opções que estavam restritas na tela de boas vindas das compilações mais

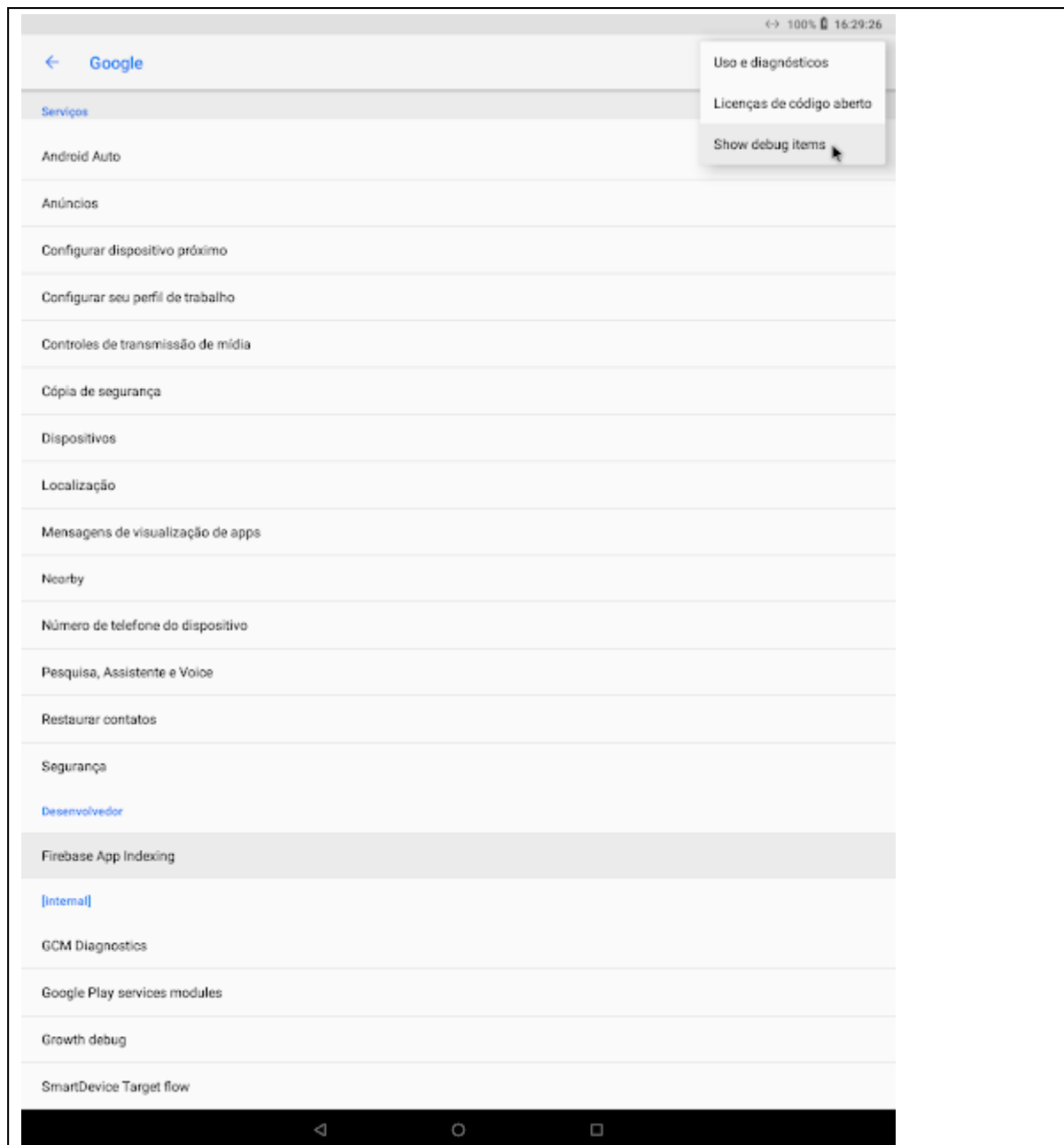
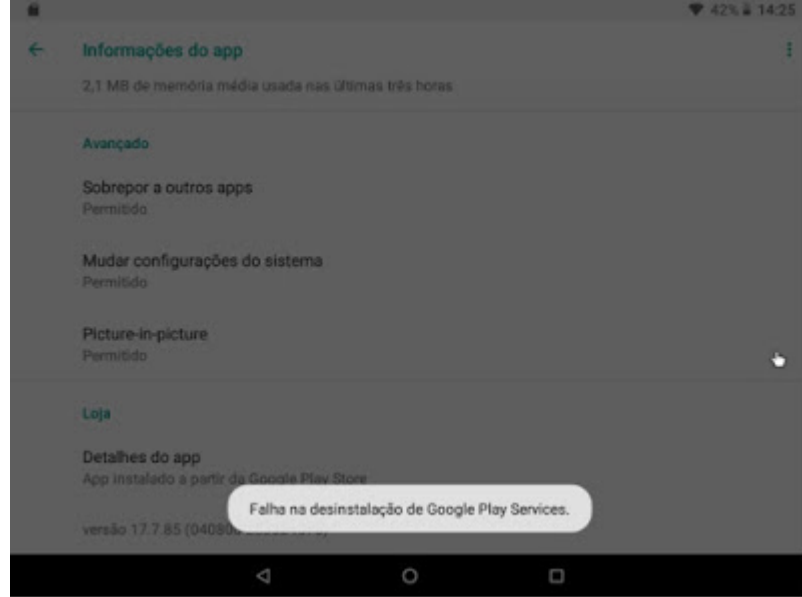
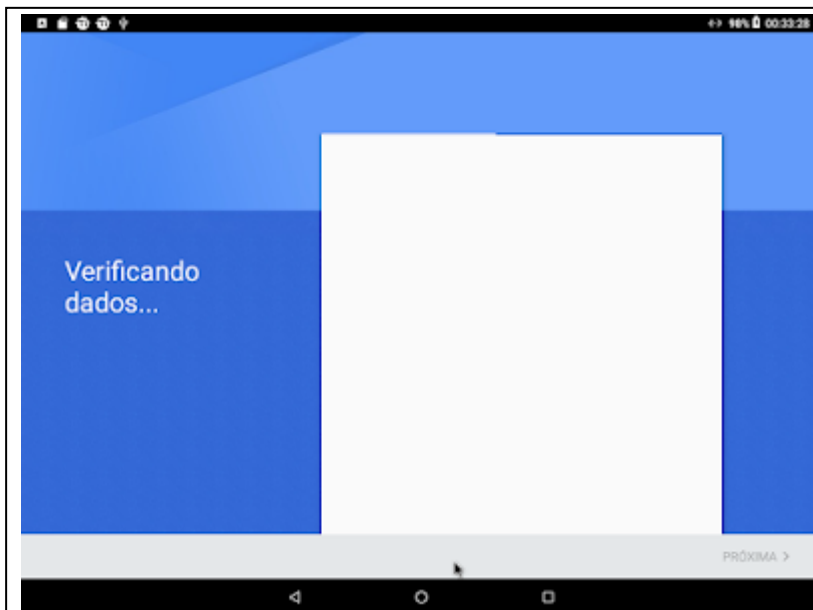
estáveis do Marshmallow. Contudo, ainda permanece somente em inglês.

A Tela Som também não mudou muito desde o Nougat, onde, apenas teve a opção Outros sons desmembrada e exibida na tela principal, além de também ter perdido a opção Transmitir.

A Tela de Armazenamento ainda segue, mais ou menos, o mesmo padrão adotado no Lollipop, cuja tela principal apenas recebeu mudanças pontuais.

Mas, ao clicar sobre o Armazenamento interno compartilhado, percebe-se aprimoramentos no seu gerenciamento, como descrições mais detalhadas e um novo recurso de limpeza, através do botão Liberar Espaço.

Ao clicar sobre o botão, abre-se a tela acima, desconectada do processo Configurações onde, no caso, não identificou nada.

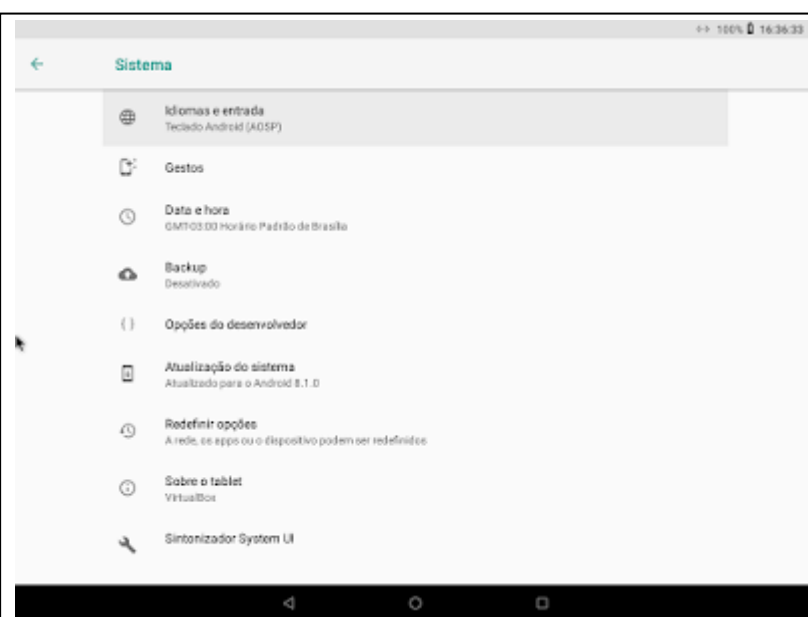
Ao selecionar a chave da opção Gerenciador de armazenamento, desativado por padrão, permite a limpeza automática de fotos e vídeos (algo só presente em *apps* de terceiros ou em ROMs customizadas, como o *app* Manutenção do Dispositivo, da Samsung).

Curiosamente, ao clicar na opção Sistema, é exibida a mensagem acima (no Nougat a mensagem era um pouco diferente: "O *sistema inclui arquivos que o Android não pode exibir individualmente.*").

A Tela Segurança foi outra que recebeu polimentos. Ganhou o resumo do Play Protect, Encontre Meu Dispositivo e Atualização de segurança, que nada são além de atalhos; teve a opção Localização integrada na guia Privacidade; perdeu a opção Fontes Desconhecidas (como citado anteriormente) e ganhou o botão Criptografia e credenciais.

O Google Play Protect, parte da Play Store, realiza a função básica dos antivírus do Android de verificar os *apps* instalados e, basicamente, monitorar o sistema. Lançado junto com o Oreo, mas presente em todas as versões suportadas do sistema, teve sua confiabilidade questionada nos primeiros meses mas está em constante aprimoramento.

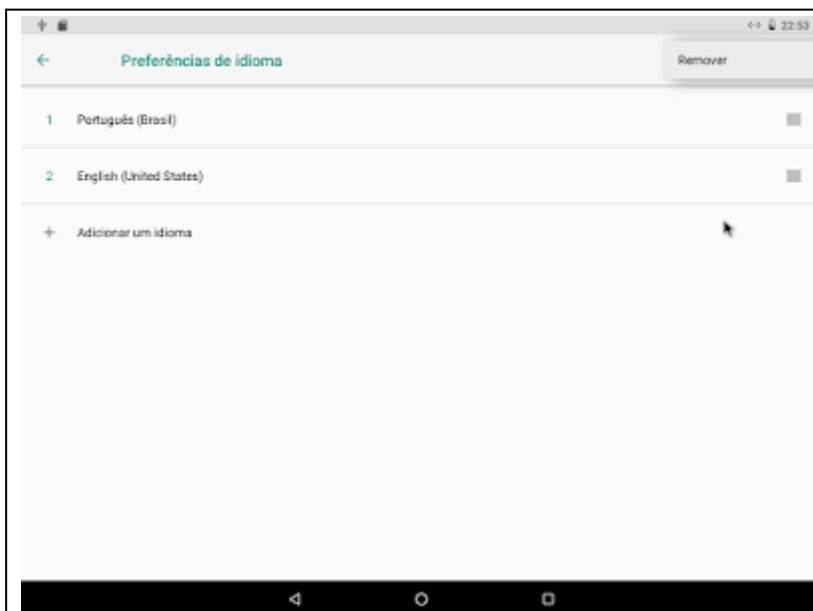
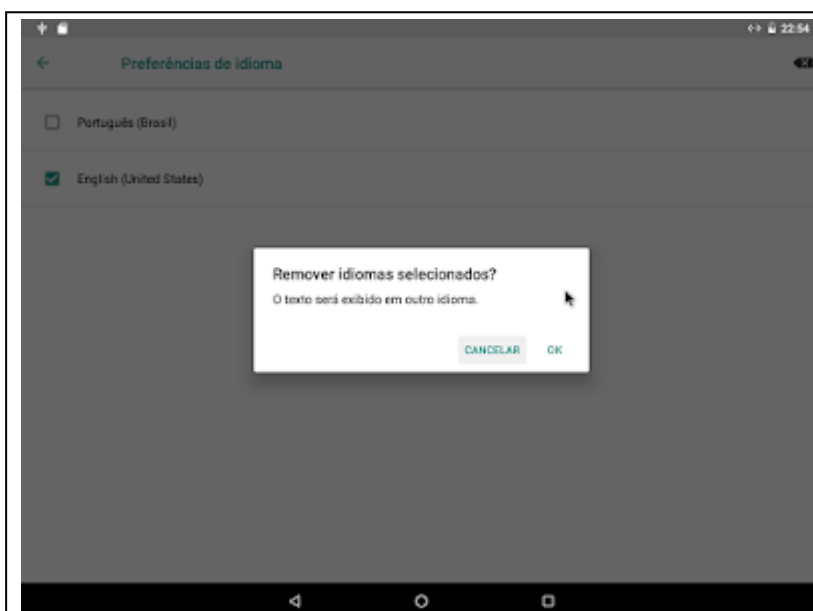
Já a tela Encontre Meu Dispositivo, exclusiva do Oreo, detalha mais a função, oferecendo mais opções para usufruir do recurso.

A tela Criptografia e credenciais reúne as opções que antes estavam na tela principal (na guia Armazenamento de credenciais).

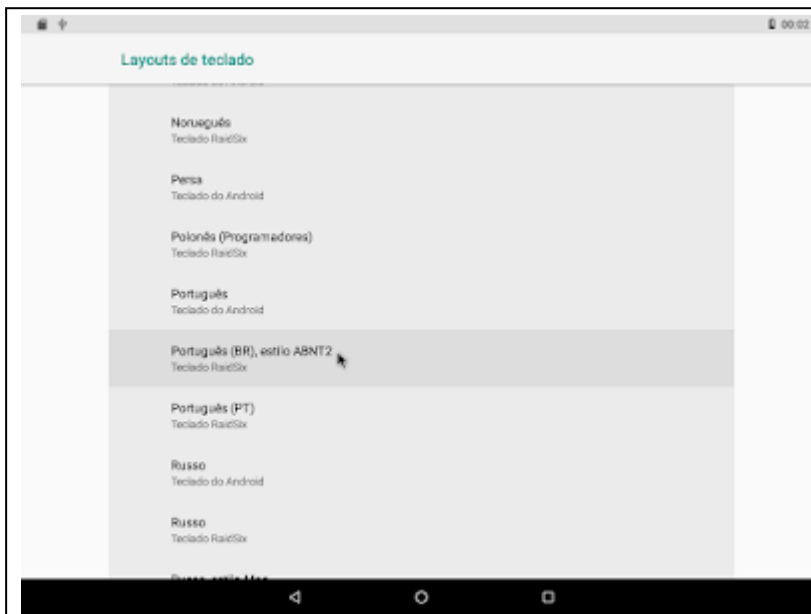
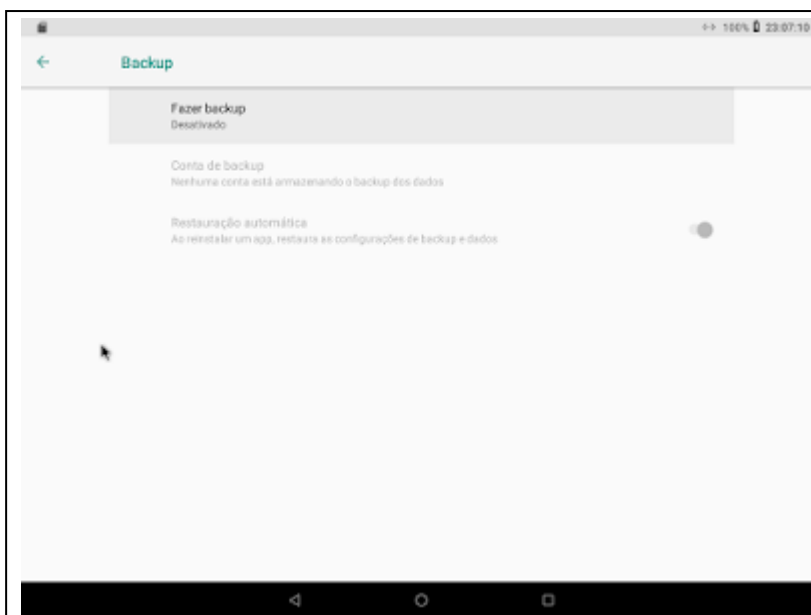
Presente desde o Nougat, o recurso Fixação de Tela não sofreu mudanças em seu funcionamento.

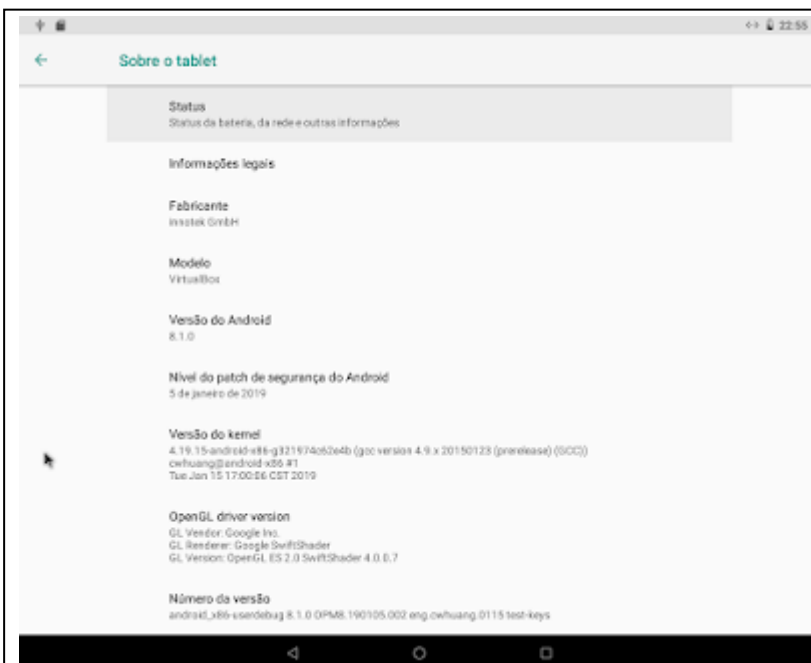
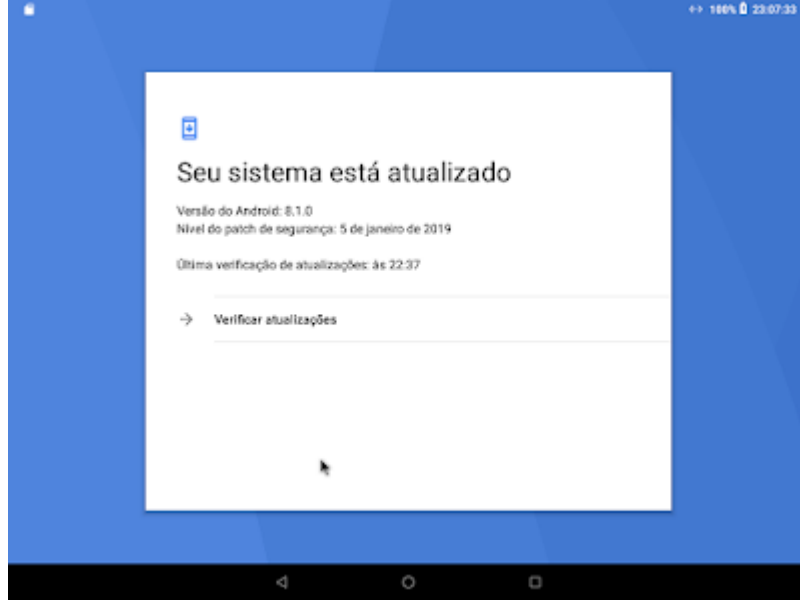
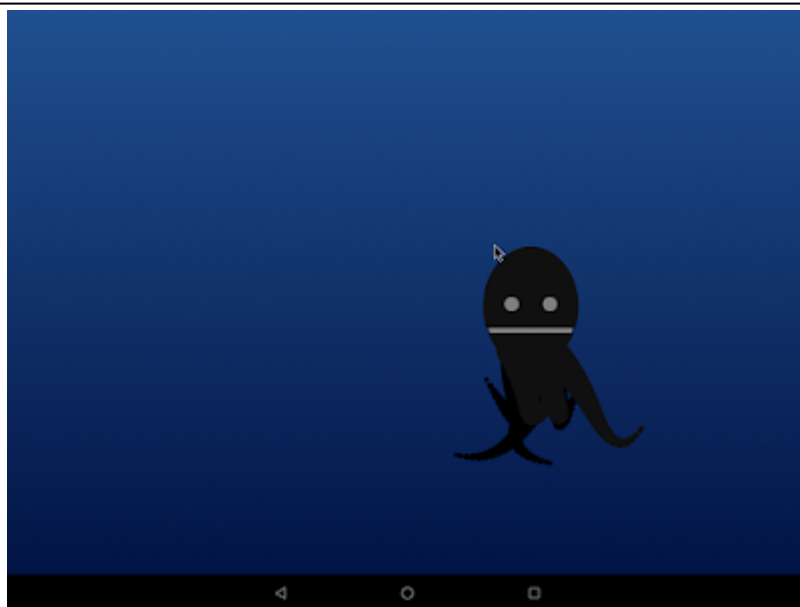
Quando habilitado, o sistema permite o bloqueio do dispositivo de acordo com o uso da função.

Prosseguindo, a tela Usuários e contas foi mais uma que recebeu aprimoramentos em relação à versão 7 do Android. A parte de usuários foi movida para uma opção própria; a opção Superusuário, antes na tela principal, também foi movida e continua redirecionando para a variação do antigo *app* do extinto projeto CyanogenMod (atual Lineage OS); recebeu as opções de sincronização de dados (antes presente na guia Rede sem fio e outras da tela principal) e ganhou a opção de adicionar usuários na tela de bloqueio (lembrando que, desde o Lollipop - Android Puro - é possível clicar no usuário no canto direito). Basicamente, o recurso sofreu aprimoramentos pela Google desde seu lançamento, no Jelly Bean, suportando melhor perfis de trabalho, para uso empresarial.

O gerenciador de Superusuário utilizado no Android x86 há muito tempo permanece não muito bem adaptado no *Material Design* do Oreo, com elementos que remetem às versões 4.x.x do sistema; contudo, ainda funciona muito melhor que a opção nativa descrita anteriormente.

A tela Acessibilidade recebeu melhorias importantes, organizando melhor suas opções, incorporando alguns recursos dispersos em Opções de

Desenvolvedor, por exemplo, estando mais completa.

A opção Google, que engloba as configurações do Play Services desde o Marshmallow, ganhou, dos tempos para cá, uma novidade: a exibição de opções de depuração, cujas opções são identificadas pela guia [Internal], ainda em inglês (o que prova que o recurso ainda está em fase de melhorias), podendo ser útil para uso e controle dos desenvolvedores.

Caso o Play Services tenha sido atualizado automaticamente e,então, realizar um upgrade do sistema x86 para x64 ou vice-versa, o sistema impedirá a desinstalação da atualização e, ao fazer isto manualmente, apagando a pasta localizada em */data/app*, o componente começará a dar erros de execução que não poderão ser atenuados, a menos que reinstale o sistema (é importante ressaltar que, num upgrade de x64 para x86, não é necessário fazer o procedimento, pois, pelo menos os Google Apps embarcados também incluem as bibliotecas de 32 bits e eventualmente serão substituídos para a versão correta após uma atualização).

Curiosidade: ao tentar autenticar pelo sistema após a instalação, poderá se deparar com o padrão de tela visto desde o Lollipop, ao invés do modelo adotado no Assistente desta versão.

Ou, com sorte, visualizará uma variação mais atualizada.

Já a tela Sistema (antes uma guia) reuniu as configurações de idioma e entrada, data e hora, redefinição e Gestos; fora a opção Sobre, Opções de Desenvolvedor e Sintonizador System UI (estas duas últimas permanecem funcionando no mesmo esquema, ocultos por padrão). Vale ressaltar que as outras opções não citadas, por alguma razão, não estavam presentes na Release Candidate 1.

A Tela Idiomas e entrada sofreu algumas modificações em relação às versões anteriores, tendo algumas opções reordenadas e modificadas, deixando a configuração mais coesa.

Agora é possível ter mais de um idioma instalado ao mesmo tempo, podendo reordenar sua posição.

Para excluir tem um passo adicional: selecionar, no canto esquerdo da tela

a opção Remover (conforme mostra a imagem acima), selecionar uma ou mais linguagens para exclusão, clicar sobre o ícone de apagar (que, sinceramente, não foi uma boa escolha, podendo causar confusão) e confirmar na caixa de mensagem (lembrando as versões mais antigas do Android).

Uma das ressalvas, em relação ao Android, é que ele ainda não suporta muito bem o uso de determinados tipos de teclado (ainda mais para um sistema que vem se adaptando, nos últimos tempos, para funcionar no modo *desktop*). Por esta razão, o *app* RaidSix Input Devices, que recomendei na análise sobre o Marshmallow, ainda é muito útil caso você queira utilizar o Android x86 com PC ou notebook, no padrão PT-BR ABNT2, por exemplo. Como o aplicativo saiu da Play Store faz algum tempo, para baixá-lo e instalá-lo de forma alternativa, clique aqui ou aqui.

A Tela Backup perdeu a opção de Configuração Original (que foi aprimorada, como será visto na próxima imagem).

A Tela Redefinir Opções (desmembrada de Backup) dá novas opções de reconfigurar o sistema. Vale ressaltar que, na compilação preliminar, somente a opção Redefinir apps estava presente.

A tela Sobre continua seguindo praticamente o mesmo padrão das últimas versões do Android, apesar de ter perdido o botão Atualização do Sistema, que foi movido. Vale ressaltar que, enquanto a RC1 vinha com o kernel 4.18, o primeiro lançamento final roda na versão 4.19, o LTS mais recente do Linux até a data da publicação deste artigo

A tela de Atualização do sistema, executado do Play Services, é a mesma do Nougat.

E, ao pressionar continuamente o *easter egg* do Oreo (primeira imagem do artigo), desta vez não teremos joguinho embutido, mas apenas um simpático polvo (ainda que seja possível brincar um pouco com ele).

E assim concluo a primeira parte desta análise. Para acompanhar a segunda parte, que trata de outros detalhes do Android Oreo, além do teste dos aplicativos e conclusão, **clique** aqui.

Ficarei aguardando os comentários, de preferência, com relação até onde foi abordado (não deixe de dar uma olhada se sua dúvida está respondida na continuação do artigo). Por ora, agradeço pela atenção e continuem ligados no Blog de Bruno A. Vieira.

(Última atualização: 26/6/2019)